



DEVELOPMENT OF ENERGY-EFFICIENT ALGORITHMS FOR WIRELESS SENSOR NETWORKS

BY

MUHAMMAD AKHLAQ

A Dissertation Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

In

COMPUTER SCIENCE AND ENGINEERING

April 2012

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This dissertation, written by *Muhammad Akhlag* under the direction of his dissertation advisor and approved by his dissertation committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY in *COMPUTER SCIENCE AND ENGINEERING*.

Dissertation Committee



Dissertation Advisor
Dr. Tarek Sheltami



Member
Prof. Dr. Shokri Selim

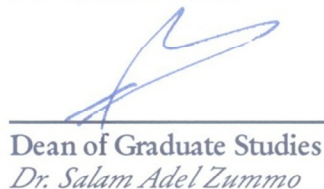


Member
Dr. Ashraf Mahmoud



Jun 29, 12

Department Chairman
Dr. Umar Al-Turki



Dean of Graduate Studies
Dr. Salam Adel Zummo



Member
Dr. Mohamed Elrabaa



Member
Dr. Mohammed Al-Sawaiyel

1/7/12

Date

Dedicated to my Parents

ACKNOWLEDGEMENT

All praises be to almighty Allah who has permitted, helped and enabled me to complete this research work; and peace and blessings of Allah be upon His final prophet Muhammad, his family and his companions.

Acknowledgment is due to the King Fahd University of Petroleum & Minerals for supporting this research. I am also thankful to King Abdulaziz City for Science and Technology (KACST) for funding this research under project # AR-29-71.

First, I must thank my supervisor, Dr. Tarek R. Sheltami, for his guidance, help, support and the enormous effort spent for me. He taught me how to think critically, write scholarly, and present clearly. I am impressed by his great sense of care, respect and flexibility for his students. Many thanks are also due to my dissertation committee members Prof. Dr. Shokri Selim, Dr. Ashraf Mahmoud, Dr. Mohamed Elrabaa and Dr. Mohammed Al-Suwaiyel for spending their valuable time and providing excellent comments in order to significantly improve this dissertation.

Second, I'm grateful to my respected teachers namely Syed Aziz-ul-Hassan Shah, Mr. Muhammad Anwar, Mr. Nazir Hussain, Syed Iqbal Hussain Shah, Mr. Mukhtar Ahmad Bhatti, Mr. Muhammad Yaqoob, Prof. Anwar Warraich, Prof. Noor Muhammad, Prof. Mehtab Anwar, Prof. Azhar H. Kazmi, Prof. Mehmood Rafiq, Prof. Hamyun Rashid, Dr. Muhammad Anwar, Dr. Abdul Rahim, Dr. Naveed A. Malik, Dr. Zubair

Nawaz, Dr. J. S. Mirza, Dr. Syed Asad Hussain, Dr. Mansoor Sarwar, Dr. Bo Helgeson, Dr. Ingvar Gustavsson, Dr. Tarek R. Sheltami, Dr. Mohammed Al-Suwaiyel, Dr. Farag Azzedin, Dr. Mohammed H. Sqalli, Dr. Mayez Al-Mouhamed, Dr. Marwan Abu-Amara, Mr. Adiche Hakim, Dr. Adel F. Ahmed, and Dr. Umar Al-Turki for their guidance and support in my academic life.

Third, I must also appreciate my family, especially my wife Sadia Bashir, my father Gulzar Ahmad, my mother Zainab Bibi, my father-in-law Muhammad Bashir Farooqi, my mother-in-law Zubaida Begum, and my brothers and sisters for their prayers, support and encouragement.

Finally, I would like to thank all friends and colleagues who helped me in any capacity during my stay at KFUPM, especially Dr. Hafiz Muhammad Afzal, Dr. Moin-ud-din, Dr. Sarfraz Abbasi, Dr. Asif Matin, Mr. Putu I. Raharja, Mr. Masud-ul-Hasan, Mr. Ahmad Tariq Sheikh, Mr. Asad Aftab, Mr. Syed Usama Idrees, Mr. M. Asif Siddiqui, Mr. Rahil Rafiq, Mr. Salman Yousaf, Mr. Saqib Sohail, Mr. Ahmad Abdul Qadeer, Mr. Zeshan Rizvi and Mr. Akhtar Hussain.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	i
TABLE OF CONTENTS	iii
LIST OF TABLES	xii
LIST OF FIGURES	xiii
NOMENCLATURE	xvi
DISSERTATION ABSTRACT	xix
ملخص بحث	xxi
CHAPTER 1. INTRODUCTION	1
1.1 Sensor devices	1
1.2 Wireless Sensor Networks (WSNs)	2
1.3 Applications of WSNs.....	3
1.4 Challenges in WSNs.....	5
1.5 Motivation	7
1.6 Problem Statement	7

1.7	Research Objectives	8
1.8	Approach or Methodology	9
1.8.1	Survey of Related Projects.....	9
1.8.2	Designing New Algorithms	9
1.8.3	Simulation of Proposed Algorithms	9
1.8.4	Mathematical Modeling	9
1.8.5	Experimental Prototype	10
1.9	Research Contributions	10
1.10	Publications	10
1.11	Organization of the Dissertation	12
CHAPTER 2.	LITERATURE REVIEW	13
2.1	Introduction	13
2.2	Duty-cycling or Sleep-scheduling.....	14
2.2.1	Contention-based protocols	15
2.2.2	Contention-free protocols	19
2.3	Topology Control	21

2.3.1	Connectivity using Power Control.....	22
2.3.2	Coverage using Power Control	24
2.3.3	Connectivity using Sleep-based methods	26
2.3.4	Coverage using Sleep-based methods.....	29
2.4	Communication Protocols	32
2.4.1	Metric-based approaches	32
2.4.2	Routing with diversity.....	33
2.4.3	Multi-path routing.....	34
2.4.4	Energy-aware routing.....	35
2.4.5	Geographic routing	36
2.4.6	Cluster-based routing	37
2.4.7	Routing to mobile sinks	39
CHAPTER 3. SAND AND DUST STORM DETECTION		41
3.1	Introduction	41
3.2	Basics of SDS Detection	44
3.2.1	Data Requirements.....	45

3.2.2	Dust Modeling	47
3.2.3	Prediction and Early Warning.....	49
3.3	Climate Implications of SDS.....	51
3.4	Ground-Based Observations	54
3.4.1	Lookout Towers	55
3.4.2	Video Surveillance.....	55
3.4.3	Sensory Information.....	56
3.5	Spaceborne Observations	58
3.5.1	Satellite Imaging	58
3.5.2	Unmanned Aerial Vehicle (UAV)	71
3.6	Hybrid Approaches	72
3.7	Comparison of Different Technologies.....	73
3.8	Conclusions	79
CHAPTER 4. CLOCK SYNCHRONIZATION		80
4.1	Introduction	81
4.2	System Model for RTSP	84

4.2.1	Clock Model.....	84
4.2.2	Assumptions.....	86
4.2.3	Time-stamping	86
4.2.4	Structure of the RTSP messages	87
4.2.5	Recursion and Multi-hop Synchronization	89
4.3	The Recursive Time Synchronization Protocol (RTSP)	90
4.3.1	Election of the Reference Node	97
4.3.2	Offset & Drift Compensation.....	100
4.3.3	Adaptive Re-Synchronization Interval	103
4.3.4	Aggregation of the Synchronization Requests.....	104
4.3.5	Security	105
4.3.6	Energy-awareness and Efficiency.....	106
4.4	Error Analysis of RTSP.....	107
4.4.1	Propagation of errors in multi-hop.....	111
4.4.2	Bounds on total error in multi-hop.....	112
4.5	Efficiency Analysis of RTSP	113

4.6	Performance Evaluation	114
4.7	Cost or Limitations of RTSP	119
4.8	Conclusions	121
CHAPTER 5. COVERAGE, CONNECTIVITY AND COMMUNICATION		122
5.1	Related Work.....	123
5.2	Basic Principals of C3	126
5.2.1	WSN as a Set of Rings.....	126
5.2.2	Triangular Tessellation Based Deployment.....	128
5.2.3	RSSI-based Distance Estimation	129
5.3	System Model for C3 Protocol.....	130
5.3.1	Assumptions.....	130
5.3.2	Distance Estimation	131
5.3.3	Structure of the C3 messages.....	131
5.3.4	Node States	132
5.4	The C3 Protocol	133
5.4.1	Formation of Rings	133

5.4.2	Formation of Clusters	135
5.4.3	Formation of Dings	137
5.4.4	Sleep-scheduling	138
5.4.5	Routing or Communication.....	138
5.5	Analysis of C3 Protocol	140
5.5.1	Connectivity	140
5.5.2	Coverage	143
5.5.3	End-to-end Delay	144
5.6	Performance Evaluation	145
5.6.1	Energy Model.....	145
5.6.2	Simulation Parameters	146
5.6.3	Results and Discussions	148
5.7	Conclusions	153
CHAPTER 6. SDS DETECTION SYSTEM		154
6.1	Introduction	154
6.2	Related Work.....	156

6.3	Challenges	158
6.4	System Architecture of SDSDS	159
6.4.1	Physical Layer.....	161
6.4.2	Adapter Layer	161
6.4.3	SDS Database.....	161
6.4.4	Context Fusion	162
6.4.5	Reasoning Engine	162
6.5	Performance of SDSDS.....	163
6.5.1	Energy Conservation.....	163
6.5.2	Efficiency and Effectiveness.....	164
6.5.3	Challenge Handling	164
6.6	Conclusions	166
CHAPTER 7. IMPLEMENTATION AND EXPERIMENTS.....		167
7.1	Experimental Setup	167
7.2	Experiments.....	170
7.2.1	Clock Synchronization using Message Exchange	171

7.2.2	Distance Estimation using RSSI	172
7.2.3	Coverage Area	173
7.2.4	Connectivity and Communication	173
7.2.5	Video Streaming over WSN	175
7.3	Infrastructure Creation using C3 Protocol	182
7.4	Clock Synchronization using RTSP Algorithm	184
7.5	SDS Detection using Experimental Prototype	185
7.6	Conclusions	186
CHAPTER 8. CONCLUSIONS AND FUTURE DIRECTIONS.....		187
8.1	Conclusions	187
8.2	Future Directions.....	190
REFERENCES		191
VITA.....		225

LIST OF TABLES

TABLE 2.1: Energy consumption in different modes [16]	14
TABLE 3.1: Region-wise sand and dust storm statistics.....	53
TABLE 3.2: Frequency of SDS worldwide for each type of land-surface	54
TABLE 3.3: Comparison of the capabilities of some satellite imaging devices ..	61
TABLE 3.4: Spectral bands used by Satellite Imagers.....	69
TABLE 3.5: Comparison of Different Technologies for SDS Detection.....	76
TABLE 4.1: Structure of an RTSP Message	88
TABLE 4.2: Simulation Parameters	115
TABLE 4.3: Comparison of RTSP with Other Algorithms.....	120
TABLE 5.1: Structure of a C3 Message	132
TABLE 5.2: Simulation Parameters for C3.....	147
TABLE 7.1: Simulation Parameters	179

LIST OF FIGURES

Figure 1.1: A Sensor and a Wireless Sensor Network [1]	2
Figure 2.1: Path-observation techniques.....	25
Figure 2.2: State diagrams for sleep-based connectivity [16].....	27
Figure 2.3: State diagrams for sleep-based coverage and connectivity [16]	30
Figure 3.1: Technologies for Sand and Dust Storm Monitoring	43
Figure 3.2: Global Map of AOD at 550 nm for April 06, 2012 00UTC.....	51
Figure 4.1: Structure of IEEE 802.15.4 MAC frame and the time-stamping	87
Figure 4.2: The request-and-reply mechanism in RTSP.....	90
Figure 4.3: Example of request-and-reply mechanism in RTSP algorithm.....	102
Figure 4.4: Computation and memory requirements of RTSP vs. FTSP.....	113
Figure 4.5: Average absolute error of time synchronization	116
Figure 4.6: Average absolute error of time synchronization (per-hop)	117
Figure 4.7: No. of nodes vs. average absolute error of time synchronization	117
Figure 4.8: Total energy consumption in the long run (for 100 nodes)	119

Figure 5.1: WSN as a set of rings (sink is at the center).....	127
Figure 5.2: Optimal deployment geometries.	128
Figure 5.3: Distance vs. LQI and RSSI values [223].....	130
Figure 5.4: RSSI, distance, and ring formation.	131
Figure 5.5: The potential forwarders or CHs in the shaded area	140
Figure 5.6: WSN divided into rings and clusters by the simulator	142
Figure 5.7: Triangular Tessellation for optimal coverage	143
Figure 5.8: Triangular Tessellation for optimal coverage within a cluster	144
Figure 5.9: R_c/R_s vs. Average Coverage Ratio	148
Figure 5.10: R_c/R_s vs. Packet Delivery Ratio	149
Figure 5.11: R_c/R_s vs. No. of Active nodes or Clusterheads (for $n=500$)	150
Figure 5.12: Communication Life Time of the System	151
Figure 5.13: Coverage Life Time of the System.....	151
Figure 5.14: No. of Nodes vs. Energy Consumption.....	152
Figure 6.1: Architecture of the proposed system (SDSDS).....	160
Figure 6.2: System Model of SDSDS	160

Figure 6.3: RGB for Monitoring of Dust Storms for Daytime & Nighttime.....	165
Figure 7.1: Experimental setup.....	168
Figure 7.2: TinyCatcher interface.....	169
Figure 7.3: SerialForwarder interface.....	170
Figure 7.4: Clock Synchronization for two nodes.....	171
Figure 7.5: RSSI vs. distance.....	172
Figure 7.6: Coverage area.....	173
Figure 7.7: No. of packets transmitted vs. throughput.....	174
Figure 7.8: No. of nodes transmitted vs. Average delay.....	175
Figure 7.9: The EvalVid Framework [231].	176
Figure 7.10: Peak signal-to-noise ratio (PSNR)	180
Figure 7.11: Throughput of the network.....	181
Figure 7.12: End-to-end delay	181
Figure 7.13: Network infrastructure created by C3 protocol.....	183
Figure 7.14: Accuracy of clock synchronization protocols.....	184

NOMENCLATURE

AFECA	Adaptive Fidelity Energy-Conserving Algorithm
AMSU	Advanced Microwave Sounding Unit
AOD	Aerosol Optical Depth
AODV	Ad-hoc On-demand Distance Vector
ASCENT	Adaptive Self-Configuring sEnsor Networks Topologies
AVHRR	Advanced Very High Resolution Radiometer
BECA	Basic Energy-Conserving Algorithm
B-MAC	Basic MAC
BTD	Brightness Temperature Difference
C3	Coverage, Connectivity and Communication protocol
CALIPSO	Cloud Aerosol Lidar and Infrared Pathfinder Satellite Observations
CBR	Constant Bit Rate
CBTC	Cone-Based Topology Control
CCP	Coverage and Connectivity Protocol
CEC	Cluster-based Energy Conservation
CLUSTERPOW	Cluster Power
COMPOW	Common Power
CPR	Cloud Profiling Radar
DESS	Delay-Efficient Sleep Scheduling
D-MAC	Data-gathering MAC
DSDV	Destination Sequenced Distance Vector
ERN	Enquiry/election of the Reference Node
ETX metric	Expected Transmission metric
ExOR	Extremely Opportunistic Routing
FTSP	Flooding Time Synchronization Protocol

GAF	Geographic Adaptive Fidelity
GEAR	Geographical & Energy-Aware Routing
GOES	Geostationary Operational Environmental Satellites
GRAB	GRAdient Broadcast
GRAd	Gradient Routing Technique
IDDI	Infrared Difference Dust Index
LDAS	Lightweight Deployment-Aware Scheduling
LDCC	Layered Diffusion-based Coverage Control
LMST	Local Minimum Spanning Tree
LQI	Link Quality Indicator
LSLR	Least Square Linear Regression
MAC	Medium Access Control
MANET	Mobile Ad hoc NETworks
MECN	Minimum Energy Connected Network
MODIS	MODerate resolution Imaging Spectroradiometer
MOR/MER	Minimum Outage Route / Minimum Energy Route
MPEG	Moving Pictures Experts Group
MPI	Microwave Polarized Index
MSG	Meteosat Second Generation
MS-MAC	Multi-parent Staggered wakeup scheduling MAC
MULE	Mobile Ubiquitous LAN Extensions
NAC	Neighbor-Aware Clusterhead
NDDI	Normalized Difference Dust Index
NOAA-POES	National Oceanic and Atmospheric Administration - Polar-orbiting Operational Environmental Satellites
NTP	Network Time Protocol
OGDC	Optimal Geographical Density Control
OMI-AI	Ozone Monitoring Instrument Aerosol Index
PAMAS	Power Aware Medium-Access with Signalling
PCA	Principal Component Analysis
PEAS	Probing Environment and Adaptive Sleep
PHAiRS	Public Health Applications in Remote Sensing

PNSR	Peak Signal-to-Noise Ratio
PSD	Particle-Size Distribution
PSO	Particle Swarm Optimization
PTP	Precision Time Protocol
RBS	Reference Broadcast Synchronization
ReSync	Reservation-based Synchronized
RSSI	Received Signal Strength Indicator
RTSP	Recursive Time Synchronization Protocol
SDS	Sand and Dust Storm
SDSDS	Sand and Dust Storm Detection System
SEAD	Scalable Energy-efficient Asynchronous Dissemination
SEVIRI	Spinning Enhanced Visible and InfraRed Imager
SFD	Start of Frame Delimiter
S-MAC	Sensor MAC
SMACS	Stationary MAC And Startup
SMECN	Small Minimum Energy Connected Network
TBF	Trajectory-Based Forwarding
TICER/RICER	Transmitter/Receiver-Initiated Cycle Receptions
TIR	Thermal Infrared
T-MAC	Timeout MAC
TOMS	Total Ozone Mapping Spectroradiometer
TPSN	Timing-sync Protocol for Sensor Networks
TRAMA	TRaffic-Adaptive Medium Access
TTDD	Two-Tier Data Dissemination
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
VIR	Visible and near-infrared
WAHSNs	Wireless Ad Hoc And Sensor Networks
WEAC	Warning Energy Aware Clusterhead
WSN	Wireless Sensor Network

DISSERTATION ABSTRACT

NAME: MUHAMMAD AKHLAQ

TITLE OF STUDY: DEVELOPMENT OF ENERGY-EFFICIENT
ALGORITHMS FOR WIRELESS SENSOR NETWORKS

MAJOR FIELD: COMPUTER SCIENCE AND ENGINEERING

DATE OF DEGREE: APRIL, 2012

Wireless sensor networks (WSNs) require very efficient algorithms for data processing, compression and routing due to the limited resources of sensor nodes. As WSNs are commonly deployed for monitoring the remote and hazardous environments, it is not feasible to recharge or replace the battery of sensor nodes. Therefore, sensor nodes should use their energy very efficiently even when a power harvesting technique is used with the battery of limited capacity. Moreover, most of the sensors' energy is consumed by data transmission and idle listening. Therefore, the lifetime of WSNs can be significantly prolonged by using energy-efficient protocols for topology control, duty-cycling, routing, and data aggregation and compression. Aiming at developing energy efficient algorithms for WSNs, this dissertation proposes several algorithms for clock synchronization, coverage, connectivity, communication and dust storm detection. First, a recursive time synchronization protocol (RTSP) for global clock synchronization in WSNs is proposed, which provides an average accuracy of $0.23\mu\text{s}$ per hop in a large multi-hop clustered network using 7-times lesser energy than that of state-of-the-art

FTSP algorithm in the long run. Second, an integrated protocol for coverage, connectivity and communication (C3) is proposed, which ensures partial coverage of more than 90% of the total deployment area, guarantees at least 1-connected network, and enables energy-efficient communication in WSNs. Finally, a sand and dust storm detection system (SDSDS) is developed, which incorporates the proposed algorithms in order to achieve high level of accuracy, reliability and energy-efficiency.

ملخص بحث

درجة الدكتوراة في الفلسفة

الاسم:	محمد اخلاق
عنوان الرسالة:	تطوير كفاءة الطاقة خوارزميات لشبكات الاستشعار اللاسلكية
التخصص:	علوم الحاسب والهندسة
تأريخ التخرج:	ابرل 2012

تتطلب شبكات الاستشعار اللاسلكية (WSNs) خوارزميات فعالة للغاية وذلك لضغط البيانات وتجهيزها وتوجيهها نظرا لمحدودية الموارد لعقد الاستشعار. وحيث أن WSNs تنتشر عادة لرصد البيئات النائية والخطرة، فإنه ليس من المجدي إعادة شحن أو استبدال البطارية من عقد الاستشعار. ولذلك، ينبغي أن تستخدم عقد الاستشعار الطاقة على نحو فعال جدا حتى عندما وجود مصادر منتجة للطاقة مع بطارية ذات قدرة محدودة. وعلاوة على ذلك، فإن معظم الطاقة تستهلك في أجهزة الاستشعار عن طريق نقل واستقبال البيانات. ولذلك، لا يمكن للحياة أن يمكن إطالة عمر WSNs بشكل كبير عن طريق استخدام بروتوكولات ذات كفاءة في استخدام الطاقة للتحكم توزيع الشبكة، والتشغيل، والتزامن، والتوجيه، وتجميع وضغط البيانات. تقترح هذه الرسالة عدة خوارزميات تهدف إلى استخدام الطاقة لـ WSNs لتطوير خوارزميات للتغطية، والربط والاتصال والكشف عن العواصف الرملية والغبار. أولا، نقترح بروتوكول ترامن متكرر (RTSP) لمزامنة ساعة الشبكة في WSNs، التي توفر دقة بمعدل 0.23 ميكرو ثانية لكل قفزة في شبكة عنقودية

واسعة متعددة القفزات باستخدام طاقة أقل 7 مرات مقارنة مع أكثر الخوارزميات المستخدمة الخوارزمية (FTSP) في المدى البعيد. ثانيًا، نقترح وضع بروتوكول متكامل للاتصال، والتغطية والربط (C3)، لكي نضمن تغطية جزئية من أكثر من 90٪ من مجموع مساحة الانتشار، وتضمن ما لا يقل عن مسار واحد للشبكة وهذا البروتوكول يستهلك الطاقة بكفاءة عالية. أخيرًا، تم تطوير نظام للكشف عن العواصف الرملية والغبار نظام (SDSDS)، الذي يتضمن الخوارزميات المقترحة من أجل تحقيق مستوى عالٍ من الموثوقية، ودقة وكفاءة استخدام الطاقة.

CHAPTER 1.

INTRODUCTION

Wireless Sensor Networks (WSNs) are one of the most important technologies of this century [1]. The advancement in micro-electro-mechanical systems (MEMS) and wireless networking technologies have made it possible to design small, inexpensive, and smart sensors. These sensors can be deployed in any environment (i.e. terrestrial/surface, underwater, underground and air) in order to collect some information, process it, and then transmit it to the sink.

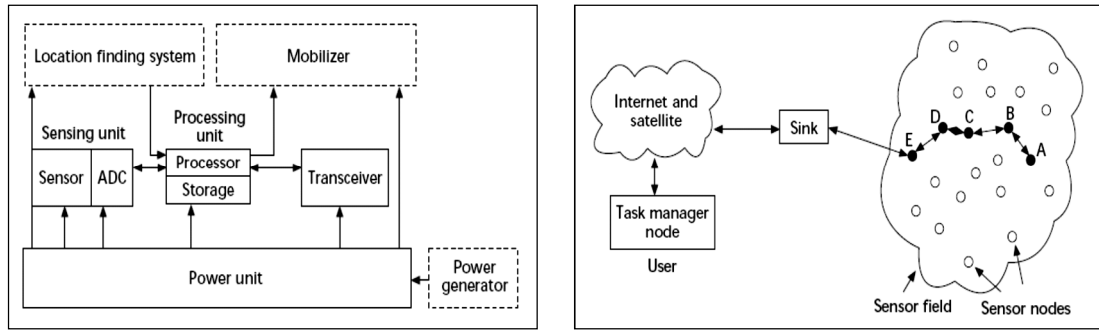
1.1 Sensor devices

A sensor has four main components [1]: a sensing unit, a processing unit with memory, a power source (usually AA batteries) , and a transceiver. Depending on the application, it can also have one or more optional components, such as an actuator, positioning system, mobilizer, and power generator (see Figure-1(a)). These resources of sensor nodes are very limited.

There are many types of sensors available nowadays, such as temperature sensor, pressure sensor, acoustic sensors, humidity sensor, optical sensors, mechanical sensors, magnetic & electromagnetic sensors, chemical & biological sensors, etc.

1.2 Wireless Sensor Networks (WSNs)

WSN is a wireless network of large number of small, cheap and low-power sensing devices densely deployed into the environment. These sensor devices collect environmental information, process it, and then transmit it to the sink that is connected to a task manager node through satellite or Internet [1] as shown in Figure-1(b).



(a) Architecture of a sensor node

(b) Architecture of a sensor node

Figure 1.1: A Sensor and a Wireless Sensor Network [1]

WSNs, based on infrastructure, have two types [2]: structured and unstructured. In a structured WSN, all or some of the nodes are placed on pre-decided locations for better coverage. This requires fewer nodes, and incurs lower cost for maintenance & management of the network. On the other hand, an unstructured WSN is the one in which sensor nodes are densely deployed into the environment in an ad hoc fashion. This network can be left unattended, but requires more nodes due to redundancy, and offers difficulties in connectivity management and failure detection.

For efficient control and management of WSNs, sensor nodes need to be self-

configured, i.e. they should configure themselves into a network soon after deployment. Due to resource limitations of the sensor nodes, efficient algorithms are required for data processing, compression, and routing in the network.

WSNs are usually deployed in remote and hazardous areas. It is, therefore, not possible to recharge the battery of these sensors, which is the main source of energy or power. Depending on the environment where a sensor will be deployed, power harvesting techniques [3–5] can be used, such as solar panels, piezoelectric fibers, and electromagnetic generators with mobile sensors, while mobile robots to recharge static sensors [6]. However, power harvesting techniques are not very common because of their immaturity, high cost, or unsuitability for certain environments. For example, solar panels are mature technology, but cannot be used for underground or underwater networks. It is important to make efficient use of sensor energy even when a power harvesting technique is used with the battery of limited capacity.

1.3 Applications of WSNs

WSNs have large number of applications that can be grouped into two types [2]: monitoring applications and tracking applications. WSNs are commonly deployed for monitoring the remote and hazardous environments, such as battle-fields, volcanoes, tsunamis, storms, seismology, etc. [7–12].

One of the pioneer WSNs was deployed for habitat monitoring of a bird called Leach on Great Duck Island, Maine [13]. This allowed researchers to observe the colony

of these birds remotely so as to avoid any kind of “observer effect”. Another important early application of WSNs was made in military for monitoring enemy’s army in the battlefield and detecting their weapons by DARPA SensIT (Sensor Information Technology) program.

Sensor networks are also being used for monitoring the health of civil structures, such as bridges and buildings [14]. They can timely alert us about crack or damage to these structures without any need of regular inspections. Current research focuses on development of controllable structures that are secure and resistant to external disturbances. A controllable structure can be developed by incorporating actuators that react to any kind of seismic waves etc.

A network of sensors and actors can be used in industrial manufacturing for process monitoring and control. The main advantage of a wireless network in such environments is its low-cost and high flexibility [14]. Similarly, WSNs have applications in refineries, pipeline operation, exploration and production, patient monitoring, retail, grocery, airports and transportations.

One of the modern applications of WSNs are in disaster management [15] that includes monitoring and/or prediction of seismic, volcano, weather, tsunamis, *dust-storms*, floods, air or water pollution, fires and dangerous gases. There are many companies producing sensor technologies for commercial applications. The development of standards, such as IEEE 802.15.4 and ZigBee, will help in developing more innovative applications in the future.

1.4 Challenges in WSNs

WSNs offer a number of interesting research challenges. Some the most important challenges include:

Extending the lifetime of WSNs: A battery-powered sensor may exhaust its energy within one month of non-stop operation. However, many applications require sensors to be alive for several months or years. The lifetime of sensor nodes can be increased by improving battery design, using energy harvesting techniques, and using energy-efficient protocols.

Effectiveness and responsiveness: The lifetime of a network can be extended by using duty cycling, i.e. periodic switching between sleep and wakeup mode. However, it may result in slow response or even miss an event. The latency must be kept within acceptable limits.

Robustness and fault-tolerance: WSNs are usually deployed in harsh environments where sensors may have high rate of failure. The protocols for WSNs must be robust to such failures and provide graceful degradation in performance.

Synergistic protocols: The synergy principal says that the capabilities of a system can be greater than the sum of the capabilities of its parts. As the capabilities of individual sensor nodes are very limited, researchers need to develop synergistic protocols for WSNs.

Scalability: There may be millions of sensor nodes in large WSNs. Therefore, WSNs should use hierarchical architecture and distributed protocols with localized communication to make it really scalable. Failures and in-situ reprogramming of sensor nodes also need to be taken care of.

Heterogeneity: A WSN may have some high-capability devices in addition to many low-capability devices. This requires us to use clustering algorithms for 2-tier architecture of the network. This heterogeneity provides multiple sensing models which require the use of sensor fusion techniques. It is an important challenge to find a right combination of these heterogeneous devices.

Self-configuration: WSNs are generally unattended because they are deployed in remote areas. It is, therefore, important that the sensor nodes organize themselves into a network soon after deployment. It also requires automatic localization, synchronization, calibration, coordination, and finding other parameters for network operations.

Optimization and adaptation: Sensor nodes should learn from the network measurements and use it for enhancing their performance over time. Also, the protocols should be able to adapt to any kind of changes in the network.

Systematic design: The sensor nodes in a WSN have very limited resources and require a very systematic design which should allow reusability, modularity, and run-time adaptation.

Security and privacy: The data collected by WSNs is usually very sensitive.

Therefore, it is a challenge to guarantee both security and privacy of this data.

Efficient usage of energy: Efficient usage of energy is an important issue in WSNs due to limited battery power of the sensor nodes. Any node that exhausts its battery dies and disconnects from the network. The lifetime of a network is determined by its connectivity and the total number of active nodes in it. It is important to note that the most of the sensors' energy is consumed in *data transmission* – transmitting the sensor data, and *idle listening* – listening to the idle channel. Therefore, efficient usage of energy can be ensured by improved topology control, duty-cycling, communication protocols, and aggregation & compression of data.

This thesis will propose and implement some energy-efficient schemes for topology control, duty-cycling and communication protocols.

1.5 Motivation

WSNs are commonly deployed for monitoring the remote and hazardous environments, such as battle-fields, volcanoes, tsunamis, storms, seismology, etc. It is, therefore, not possible to recharge the battery of these sensors, which is the main source of energy or power. It is important to make efficient use of sensor energy even when a power harvesting technique is used with the battery of limited capacity.

1.6 Problem Statement

Efficient usage of energy is a key issue in WSNs due to limited battery power. A

sensor node that exhausts its battery dies and disconnects from the network reducing the lifetime of a network. Most of the sensors' energy is consumed in data transmission, and idle listening. Therefore, efficient usage of energy could be ensured by improved topology control, duty-cycling, communication protocols, and aggregation & compression of data.

1.7 Research Objectives

The proposed research is a part of KACST project which aims at designing and implementing WSN for monitoring road conditions and detecting sandstorms and providing early warnings with detailed information to all drivers on the roads.

Following are the main objective of this research:

1. Developing a novel and efficient *sleep-scheduling algorithms* for wireless sensor networks.
2. Developing an energy-efficient *topology control scheme* for wireless sensor networks.
3. Developing *efficient and power conservative routing protocols* and integrating those into the sensor nodes.
4. Establishing and promoting a smart wireless sensor network for *detection and prediction of sandstorms and initiating warnings* to the drivers. The proposed energy efficient algorithm will be incorporated in this application.

1.8 Approach or Methodology

To achieve the objectives of this research, following approach are used:

1.8.1 Survey of Related Projects

A survey of the recent literature is done to identify the requirements for designing a wireless sensor network for detection of sandstorms.

1.8.2 Designing New Algorithms

New algorithms related to duty-cycling, topology control, routing, and sandstorm detection are proposed. The efficient usage of energy is the prime feature of these algorithms.

1.8.3 Simulation of Proposed Algorithms

The proposed algorithms are tested by performing simulations using network simulators, such as NS2, MATLAB, Prowler, etc. Another research group provided the required data for simulations. The results obtained from these simulations were used to modify and adapt the proposed algorithms for optimal performance in our target WSN.

1.8.4 Mathematical Modeling

A theoretical analysis of the proposed algorithms is done and mathematical models are developed.

1.8.5 Experimental Prototype

An experimental prototype of a WSN for sandstorm prediction and detection is developed. To test the energy-efficiency of proposed algorithms, they were incorporated into the prototype.

1.9 Research Contributions

This dissertation proposes several energy-efficient algorithms for clock synchronization, coverage, connectivity, communication, and dust storm detection.

1. **FTSP protocol:** a recursive time synchronization protocol (RTSP) for global clock synchronization in WSNs is proposed, which provides an average accuracy of $0.23\mu\text{s}$ per hop in a large multi-hop clustered network using 7-times lesser energy than that of FTSP in the long run.
2. **C3 protocol:** a protocol for coverage, connectivity and communication (C3) is proposed that ensures partial coverage, connectivity and energy efficient communication in wireless sensor networks.
3. **SDSDS system:** a layered-architecture of the sand and dust storm detection system (SDSDS) is proposed which incorporates our proposed algorithms in order to achieve high level of accuracy, reliability and energy-efficiency.

1.10 Publications

Based on research in this dissertation, following journal and conference papers have been produced:

Journal Papers

- **Muhammad Akhlaq**, Tarek R. Sheltami, Hussein T. Mouftah, “A Review of Technologies for Sand and Dust Storm Detection”, Reviews in Environmental Science and Biotechnology, May 29, 2012 [Online First: <http://www.springerlink.com/content/u5736942j288v657/>].
- **M. Akhlaq**, Tarek R. Sheltami, Bo Helgeson and Elhadi M. Shakshuki, “Designing an Integrated Driver Assistance System Using Image Sensors”, Journal of Intelligent Manufacturing, pp. 1-24, January 03, 2012, [Online First: <http://dx.doi.org/10.1007/s10845-011-0618-1>].
- **M. Akhlaq**, Tarek R. Sheltami, “RTSP: An Accurate and Energy-efficient Protocol for Clock Synchronization in WSNs”, IEEE Transactions on Instrumentation & Measurement [Submitted; Manuscript# IM-12-5717].

Conference Papers

- **M. Akhlaq**, Tarek R. Sheltami, “Performance Comparison of Video Compression and Streaming over Wireless Ad Hoc and Sensor Networks using MPEG-4 and H.264”, in Proceedings of the Fourth International Conference on Networked Digital Technologies (NDT 2012), Part II, CCIS 294 R. Benlamri (Ed.), pp. 369--378. Springer, Heidelberg. April 24-26 2012.
- **M. Akhlaq**, Tarek R. Sheltami, “The Recursive Time Synchronization Protocol for Wireless Sensor Networks”, in Proc. of 2012 IEEE Sensors Applications Symposium (SAS 2012), Feb 7-9, 2012, Italy, pp. 62-67. [<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6166318>]
- **M. Akhlaq**, “Performance Comparison of MPEG-4 and H.264 Video Coding Standards over Wireless Ad Hoc Networks”, In Proceedings of Second Scientific Conference, Jeddah, Saudi Arabia, March 28-31, 2011.
- **M. Akhlaq**, “Introducing Context-awareness in Operating Systems for PCs: User Interface Adaptation”, In Proceedings of Second Scientific Conference, Jeddah, Saudi Arabia, March 28-31, 2011.

- **M. Akhlaq**, Tarek R. Sheltami, “*Sand and Dust Storm Detection System using Wireless Sensor Network and Satellite Imaging*”, 14th International Conference on Information Integration and Web-based Applications & Services (iiWAS 2012) [Submitted].

1.11 Organization of the Dissertation

The rest of this dissertation is organized as follows. Chapter 2 provides a comprehensive review of the literature regarding energy conservation techniques based on topology control, sleep-scheduling, and routing or communication. The techniques and technologies for sand and dust storm detection are discussed in chapter 3. In chapter 4, the recursive time synchronization protocol (RTSP) for WSNs is proposed. An integrated protocol for coverage, connectivity and communication (C3) is proposed in chapter 5. The architecture of sand and dust storm detection system (SDSDS) is proposed in chapter 6. Chapter 7 describes the experimental prototype which is established in order to implement and experiment with the proposed algorithms. Finally, chapter 8 concludes the dissertation and presents future directions.

CHAPTER 2.

LITERATURE REVIEW

At the network layer of WSNs, energy can be conserved through topology control, sleep-scheduling or duty-cycling, routing or communication protocols, and data aggregation. This chapter provides a comprehensive review of the commonly used energy conservation techniques which are based on the first three methods.

2.1 Introduction

Nodes in WSNs may be in either sleep mode or active mode. A node must be in active mode before it can receive or transmit a message. It is important to note that the sleep mode usually consumes three orders of magnitude less energy, while energy costs for receive and transmit are of the same magnitude as shown in Table 2.1 [16]. The energy costs for idle listening, receiving, and overhearing are almost similar (within 20-30%). Therefore, it is best to keep the sensor radio in *sleep mode* when inactive. Another important factor in energy consumption is the *radio transmission power*. An increase in radio transmission power increases the communication range, connectivity, and link quality. However, it also increases the energy consumption and interference. This gives us an opportunity to conserve energy by minimizing the number of transmissions and by putting the sensor nodes to sleep mode if redundant or have no data to send or receive.

TABLE 2.1: Energy consumption in different modes [16]

Radio	Frequency/ Data- Rate	Startup Time	Sleep	Receive	Transmit
CC2420	2.4GHz,250kbps	0.6ms	60 μ W	59mW	52mW
CC1000	868MHz,19.2kbps	2.0ms	0.6 μ W	29mW	50mW
MIT μ AMPS-1	2.4GHz,1Mbps	0.5ms	negligible	279mW	330mW
IEEE802.11b	2.4GHz,11Mbps	1.0ms	negligible	1.4W	2.25W

Section 2.2 describes contention-free and contention-based protocols for duty-cycling or sleep-scheduling. Section 2.3 provides an overview of the coverage and connectivity protocols using power control methods and sleep-based methods. Finally, section 2.4 describes communication protocols of several types including metric-based approaches, routing with diversity, multi-path routing, energy-aware routing, geographic routing, cluster-based routing, and routing to mobile sinks.

2.2 Duty-cycling or Sleep-scheduling

Duty cycling relates to the fraction of time a device or system is in active state. The duty-cycling is achieved by putting the radio of a node to sleep mode when inactive. Traditional media access control (MAC) protocols try to provide efficient utilization of

the shared medium in terms of *throughput, delay, and fairness*. However, MAC protocols for WSNs perform an additional function of *energy-efficiency* through duty cycling.

Traditional MAC protocols, such as Aloha, CSMA, MACA, IEEE 802.11, and IEEE 802.15.4 will not be described here because they do not explicitly consider energy efficiency.

2.2.1 Contention-based protocols

In *IEEE 802.11 infrastructure mode*, nodes can go to sleep mode by informing access point (AP). The AP will buffer all messages for sleeping nodes and will deliver to them when they wakeup periodically. It saves energy but at the cost of throughput and delay.

The *Power Aware Medium-Access with Signalling (PAMAS)* [17] protocol is an extension of MACA protocol. In PAMAS, a node that wants to send some data broadcasts a request to send (RTS) signal, and the target node broadcasts clear to send (CTS) signal. The RTS/CTS signals are sent on separate channel. A node goes to sleep mode whenever it fails to send/receive data, or when its neighbor is transmitting/receiving data to/from another node. This protocol saves on overhearing and is one of the first energy-aware protocols. However, *idle reception* can still result in wastage of energy.

Further energy can be saved by *reducing the idle reception*. This is done by putting nodes to sleep mode mostly, but awake them before receiving any packet

intended for them. There are two classes of such protocols: asynchronous, and synchronous.

Asynchronous sleep techniques keep nodes in *sleep mode as default* and wake-up only for short time to send/receive data or check for network traffic. They rely on an additional radio for wake-up signals or periodic low-power listening to wake up the receiver before any incoming transmission.

The *secondary wake-up radio* [18] method is hardware-solution that requires two radios: data radio, and wake-up radio. Nodes may sleep to save energy whenever they are idle. However, a low-power secondary radio is always on so that the nodes can receive wake-up signal from other nodes before data transmission. The main problem with this method is that it will wake-up all the nodes within broadcast range.

In *low-power listening/preamble sampling protocol* [19], [20], nodes periodically wakeup to listen the channel. If no activity is found, they sleep again. A node that wants to transmit data sends a wake-up/preamble signal (i.e. a long RF pulse) before sending data. The receiving node will go to full active mode on detecting such signal and receive the data. This technique may also wake-up all the nodes within broadcast range.

The *WiseMAC protocol* [21] avoids the long RF pulse used in above technique. Each node learns periodic sampling time of its nearby nodes from ACK packets, and sends a shorter RF pulse at an appropriate time. The WiseMAC packet uses “more” bit to inform the receiver to stay wake-up to receive more data.

The *Transmitter/Receiver-Initiated Cycle Receptions (TICER/RICER)*[22] are similar to low-power listening/preamble sampling. In TICER, transmitter node periodically sends a sequence of interrupted RTS signals whereas receiver node wakes-up periodically to sense the channel for any such signals. The receiver replies with CTS signal on detecting RTS signal and stays wake-up to receive data. In RICER, the receiver node periodically wakes-up to run three steps: monitor, send wake-up beacon, and monitor. The node that wants to transmit data will have to wait for wake-up beacon from receiver before transmission. The TICER/RICER techniques are easier to implement at higher layers than at lower layers.

The *Basic MAC protocol (B-MAC)*[23] is lightweight and highly reconfigurable protocol with three main features: Low-power listening (LPL) – chose channel sampling durations and preamble durations, Clear channel assessment (CCA) – checks if the channel is clear, and Acknowledgements (ACK) – facilitate response for any unicast packet. The B-MAC protocol is designed using modular approach and provides all the core functionality for higher layers. This protocol has been used as base protocol in many other protocols.

On the other hand, the *Synchronous sleep techniques* use *periodic duty-cycled sleep schedules* for nodes so that transmitters anticipate when their intended receivers will be awake.

The *Sensor MAC (S-MAC)* protocol [24], [25], designed for WSNs, uses a periodic sleep-listen schedule where each nodes sleeps for some time and then wakes-up

for some time to sense the channel. Any node, called synchronizer node, can take initiative to make and broadcast this schedule. In this way, there is only one sleep-listen schedule for all nodes. S-MAC uses RTS/CTS signals, and interfering nodes are put to sleep mode. It ensures energy conservation but at the cost of increased delay due to sleep latency.

The ***Timeout MAC (T-MAC)*** [26] uses fixed-length duty cycles but the end of an active period is decided by timeout. After receiving a message, the receiver node starts a fresh timer. If any message is received during this period, it resets the timer; otherwise it goes to sleep mode. T-MAC suffers from early sleep problem due to contention on sender side. However, this problem can be solved in two ways: use a short FRTS (future request to send) control message to tell the recipient to extend its timeout period, or use “full buffer priority” where a node prefers sending over receiving when its buffer is full. T-MAC also ensures energy conservation but at the cost of increased delay due to sleep latency.

The ***Data-gathering MAC (D-MAC)*** [27] avoids data-forwarding interruption problem as caused by sleeping nodes in above schemes. D-MAC applies receive–transmit–sleep sequence only on flows on a predetermined data-gathering tree rooted on sink. It minimizes delays by ensuring that when a node at level k is transmitting, the node in the upper level $k+1$ is receiving. D-MAC has many advantages: active periods may be adaptive; interference may be reduced due to active periods at different levels; fewer nodes are required to be active; and it uses data prediction and more-to-send (MTS) for

better performance. The main problem with this scheme is that it can be used with only one-way data gathering trees.

The *Delay-Efficient Sleep Scheduling (DESS)* [28] is a combinatorial optimization problem where each node selects a unique reception slot out of k slots and tells its neighbors that it will receive only in that slot. DESS is an NP-hard problem which tries to minimize path delay that is equal to the sum of all the per-hop delays.

The *Asynchronous sleep schedules* [29] do not require any inter-node synchronization and ensure that the wake-up periods for neighbors overlap. It can be formulated as a combinatorial problem of designing wake-up schedule function (WSF) for all nodes such that wake-up periods for neighboring nodes overlap at least m times.

2.2.2 Contention-free protocols

Contention-free protocols ensure that no nodes within two hops of each other may utilize the same slot. They use techniques such as time division multiple-access (TDMA), frequency division multiple-access (FDMA), or code division multiple-access (CDMA) etc. A node can sleep to save energy whenever it is not scheduled for communication.

The *Stationary MAC And Startup (SMACS)* [30] provides a decentralized approach where each node is required to have local synchronization. This is done by selecting a common communication slot with a neighbor on startup. In SMACS, each link uses randomly chosen frequency or CDMA frequency-hopping code to ensure no contention.

The *BFS/DFS-based scheduling* assigns channels on the bases of Breadth-First Search (BFS) or Depth-First Search (DFS) traversals of data-gathering trees. Each node is assigned channel slots according to its load. BFS allocates contiguous time slots to each node whereas DFS allocates non-contiguous time slots. The drawback of this technique is that it needs global synchronization.

The *Reservation-based Synchronized (ReSync)* MAC protocol [31] is flexible in terms of letting the traffic from each node to vary over time. In ReSync, nodes send a short intent message before transmitting, and also learn when their neighbors send such intent messages to wake-up in time to listen to them. When a node wants to transmit data, it sends a short intent message containing incremental data transmission time selected at random so that the intended receiver must be wake at that time. Any collisions are not persistent because data transmission times are selected at random.

The *TDMA-based MAC protocols* are nice alternative to above protocols. They refrain from idle listening through pre-scheduled communication and hence give energy efficiency. However, they use complex distributed algorithms and require tight synchronization. One example of such techniques is *Traffic-Adaptive Medium Access (TRAMA)* protocol [32]. It consists of three main parts: neighbor protocol (NP) to exchange one-hop neighboring info, schedule exchange protocol (SEP) to publish nodes' schedule, and adaptive election algorithm (AEA) to allow only highest priority node within two-hop region to transmit. All the scheduled transmitter/receivers must stay awake, while other nodes can sleep to save energy. TRAMA also provides periodic

rescheduling to handle dynamic traffic conditions. The main advantage of TRAMA is that there is no packet-loss due to collision or due to transmission to a node in sleep mode.

2.3 Topology Control

Sensor nodes are required to self-configure themselves into a network topology soon after deployment. This initial topology may change with the passage of time due to a number of factors, including node failure, energy depletion, and mobility. The process of configuring and re-configuring of the topology after deployment is known as topology control. Sensors in WSNs can be deployed in one of the two ways: random and structured (i.e. pre-planned). However, the communication network can be setup into several different topologies, such as star, mesh, grid, and hierarchical [16].

The main objective of the network is to obtain desired level of coverage and connectivity. Coverage is related to the number of sensor nodes covering any area in the network; and it affects the quality of information received. Connectivity is concerned with the degree to which a network has connections; and it affects routing of information.

The topology of a network is affected by coverage and connectivity, which in turn are affected by the radio transmission power and the number of sleeping nodes in a network.

2.3.1 Connectivity using Power Control

The power-based topology control is an optimization approach which aims at either minimizing the total power consumption over routing paths or minimizing the transmission power settings of each node.

The *Minimum Energy Connected Network (MECN)* protocol [33] derives a minimum power topology (i.e. an optimal spanning tree rooted at sink). A graph topology is called as minimum power topology, if for any pair of nodes there exists a path in the graph that consumes the smallest amount of energy compared with any other possible path. Each *node's enclosure* is described as area around it, such that direct communication is energy efficient than relay in that area. The *enclosure graph* is the one that includes all links between every node and its adjacent nodes in the related enclosure region. The MECN algorithm first makes an enclosure graph in distributed way, and then prunes it using a link energy cost-based Bellman-Ford algorithm to find the minimum power topology. The main problem with MECN algorithm is that it may not produce a connected topology with the least number of edges, i.e. there can be redundant edges. There is an improvement over MECN called *Small Minimum Energy Connected Network (SMECN)* that produces a smaller minimum power topology compared to MECN.

The *Common Power (COMPOW)* protocol [34] guarantees that the minimum common power level, which makes sure maximum network connectivity, is chosen by all the sensor nodes. A low common power level has many advantages, including symmetric

received signal power on every link; close to the best network capacity; low-power routes; and minimum contention. The COMPOW protocol first runs shortest path algorithm (e.g. Bellman-Ford algorithm) several times one for each possible power level, and then each node sees the routing tables made by algorithm and selects the lowest power level having maximum number of reachable nodes. The main drawbacks of COMPOW are that it is not very much scalable, and that one isolated node can cause all other nodes to select higher power level.

The *Minimizing Maximum Power* [35] is another protocol that can generate a connected topology (and also bi-connected topology) with non-uniform power levels by minimizing the maximum power level among all nodes. This approach seeks to minimize the energy burden on most loaded node.

The *Cone-Based Topology Control (CBTC)* method [36], [37] gives a direction-based distributed rule to guarantee a connected network topology with minimal power usage for each node. In CBTC, a node continues to increase its transmission power until it finds at least one neighbor in every cone of angle α or it attains its maximum transmission power limit. CBTC assumes that communication range increases monotonically with transmit power. We will always get a connected network if $\alpha \leq 2\pi/3$ [36] or $\alpha \leq 5\pi/6$ for tighter limits [21] or $\alpha = \pi$ when there is no limit on maximum power.

The *Local Minimum Spanning Tree (LMST)* [38] makes a consistent global spanning tree topology in a fully distributed way. Each node first makes a local minimum spanning tree (LMST) for visible portion of the graph, i.e. within max power range. This

local graph is then modified with proper weights for uniqueness. This results in a connected network topology. Simulations show that, in terms of average node degree, LMST performs better than both MECN and CBTC.

2.3.2 Coverage using Power Control

Connectivity metrics are usually application independent, whereas coverage metrics are highly dependent on application. There are mainly two coverage metrics: *k*-coverage and path-observability.

The *K-coverage* metrics determines the degree of sensor coverage overlap. A region is called to be *K*-covered if every point in that region is also in at least *K* node regions. The relationship between *K*-coverage and *K*-connectivity is that *K-coverage implies K-connectivity if $R_C \geq 2R_S$* [39], where R_C is communication range and R_S is sensing range. Note that communication range increases monotonically with transmit power.

The *path-observability* metrics are suitable for object tracking applications. Some nice examples of such metrics are maximal breach distance metric using Voronoi cell edges, and maximal support distance using Delaunay triangulation edges [40] as shown in Figure 2.1.

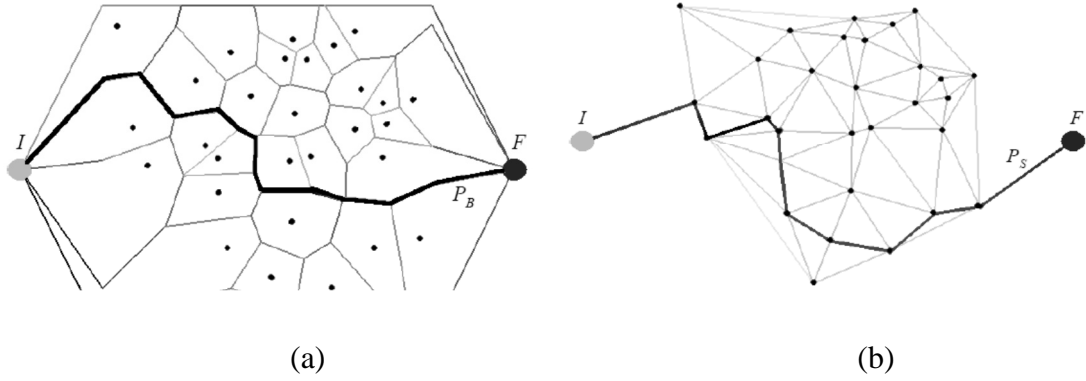


Figure 2.1: Path-observation techniques

(a) Maximal breach path using Voronoi cell edges and (b) Maximal support path using Delaunay triangulation edges. [40]

The highlighted path in Figure-2.1(a) depicts a worst-case observability of a path taken by a moving object, i.e. it has lowest probability of detection, whereas Figure 2.1(b) shows a best-case observability of a path taken by a moving object. It is important to note that, energy consumption in moving a node to some un-covered area should be minimized.

Some *other metrics* include percentage of desired points covered in the region, total area of the K-coverage, average coverage overlap, inter-node distance, and probability of target detection [16].

The coverage and connectivity of WSNs can be altered by node mobility. Therefore, it is important to briefly describe methods of *deployment with mobile nodes*. A fully distributed and localized technique for non-overlapping coverage is the potential

field technique in which mobile nodes spread out in an area by using virtual repulsive forces [41]. There is another similar technique known as Distributed Self-Spreading Algorithm (DSSA) [42]. An incremental self-deployment algorithm (ISDA) [43] calculates new location for placement based on current deployment, a new node is placed there and other nodes move to fill any gap. In a bidding protocol [44], initially static nodes are deployed which determine any coverage holes, and then mobile nodes move into these coverage holes. In another technique [45], a mobile robotic node explores an area to find where static nodes should be deployed, and then these static nodes (after deployment) guide robotic node in further exploration. In helicopter-based deployment [46], static nodes are dropped from helicopter, and are self-configured for connectivity. In case of disconnected network, the helicopter is told about where to deploy more nodes.

2.3.3 Connectivity using Sleep-based methods

In many applications using cheap sensors, it is very common to over-deploy the network with higher density or redundancy. The lifetime of over-deployed WNSs can be extended by keeping redundant nodes sleep or inactive until some neighboring node fails or depletes its energy. Sleep-based topology control techniques aim at finding which sensor nodes need to be active at a given time. Several techniques have been proposed for sleep-based topology control. These techniques mostly work in a similar way: each node uses some local rules and observations to transition between sleep, test, and active modes. Figure 2.2 shows the state diagrams for popular sleep-based topology control for connectivity only.

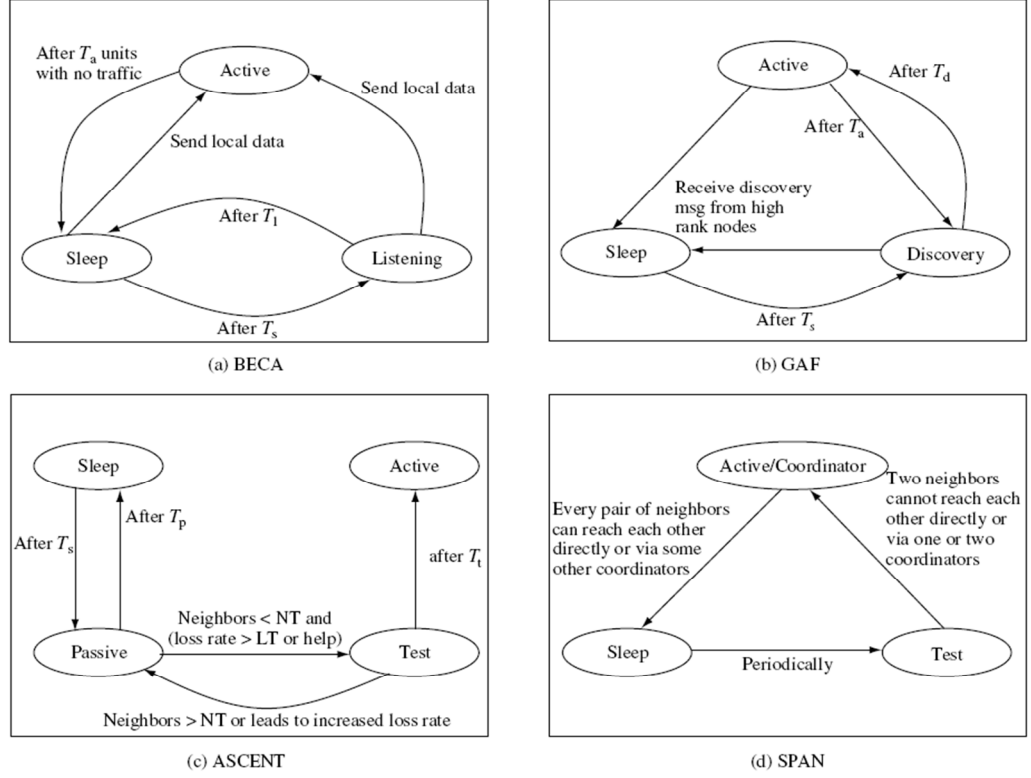


Figure 2.2: State diagrams for sleep-based connectivity [16]

The Basic Energy-Conserving Algorithm (BECA) [47], as shown in Figure 2.2(a), uses three states: sleep, listen, and active. A sleeping node will go to listening state after expiry of timer T_s to check if there is some data for it. A listening node will go to active state if it has some data to route, or otherwise return to sleep state again after waiting for T_l . A sleeping node may also directly go to active state if it has some data to send. An active node will remain active until it has some data to send or receive, or otherwise go to sleep state after waiting for T_a . In other words, a node will be in active state only when it has to send or receive some data, or participate in routing. An extend version of BECA is the Adaptive Fidelity Energy-Conserving Algorithm (AFECA) [47] that modifies sleep

time depending on the number of neighbors (N). It calculates sleep time as basic sleep time multiplied by a random number between 1 and N, i.e. $T_s = T_s * \text{rand}(N)$.

The Geographic Adaptive Fidelity (GAF) [48], as explained in Figure 2.2(b), uses three states: sleep, discovery, and active. It creates virtual square grids in which side length of each grid cell r is so small that $r < \sqrt{R/5}$, where R is the communication range. The nodes in discovery or active state broadcast discovery messages containing their ID, grid ID, and remaining lifetime. A node will go to sleep state if it hears discovery message from higher priority node in its grid. The objective of GAF is to ensure that only one node with the highest remaining lifetime is active in every virtual grid. A similar technique is the Cluster-based Energy Conservation (CEC) algorithm [33] that, instead of virtual grids, creates a set of clusters each having an elected cluster-heads with highest remaining lifetime. It also elects some gateway nodes to ensure connectivity of the network. Any node may go to sleep state for specified time if it is not a cluster-head or gateway node. All the sleeping nodes wake up periodically to re-run election algorithm.

The Adaptive Self-Configuring sEnsor Networks Topologies (ASCENT) [49], as explained in Figure 2.2(c), uses four states: sleep, passive, test, and active. The nodes wakeup to help in routing if the number of active neighbors is lesser than desired threshold and either the loss rate is greater than acceptable threshold in their neighborhood or a help message is received. A notable point in ASCENT is that once a node goes to active node, it will remain active until it dies due to energy depletion.

The SPAN protocol [50] was designed to save power in ad-hoc networks. It keeps only coordinator nodes active at any time and connectivity with sufficient redundancy. As explained in Figure 2.2(d), nodes in sleep state periodically wakeup to go to test state to send HELLO messages containing their current coordinators and neighbors. If two neighbors are unable to reach each other directly or through one or two other coordinators, then the testing node becomes an active coordinator. Conversely, any active coordinator goes to sleep state if every pair of its neighbors can reach each other directly or through some other coordinators. SPAN uses randomized prioritized backoff to avoid multiple neighboring nodes become coordinators simultaneously. A coordinator may withdraw after some time to ensure load balancing.

2.3.4 Coverage using Sleep-based methods

The topology control involves both connectivity and coverage. This section presents some sleep-based topology control techniques for coverage (and connectivity). These techniques are almost similar to previous techniques except they activate nodes using coverage-based eligibility rule.

The Probing Environment and Adaptive Sleep (PEAS) [51] method provides topology control in highly dynamic networks. As explained in Figure 2.3(a), PEAS uses three states: sleep, probe, and active. A node in sleep state goes to probe state after a random sleep time, sends a PROBE message, and waits for a REPLY message. The node goes to active state if there is no REPLY message received; and will stay active till power

depletion or failure. The PEAS technique provides both coverage and connectivity by ensuring at least one active node within a configurable probing range.

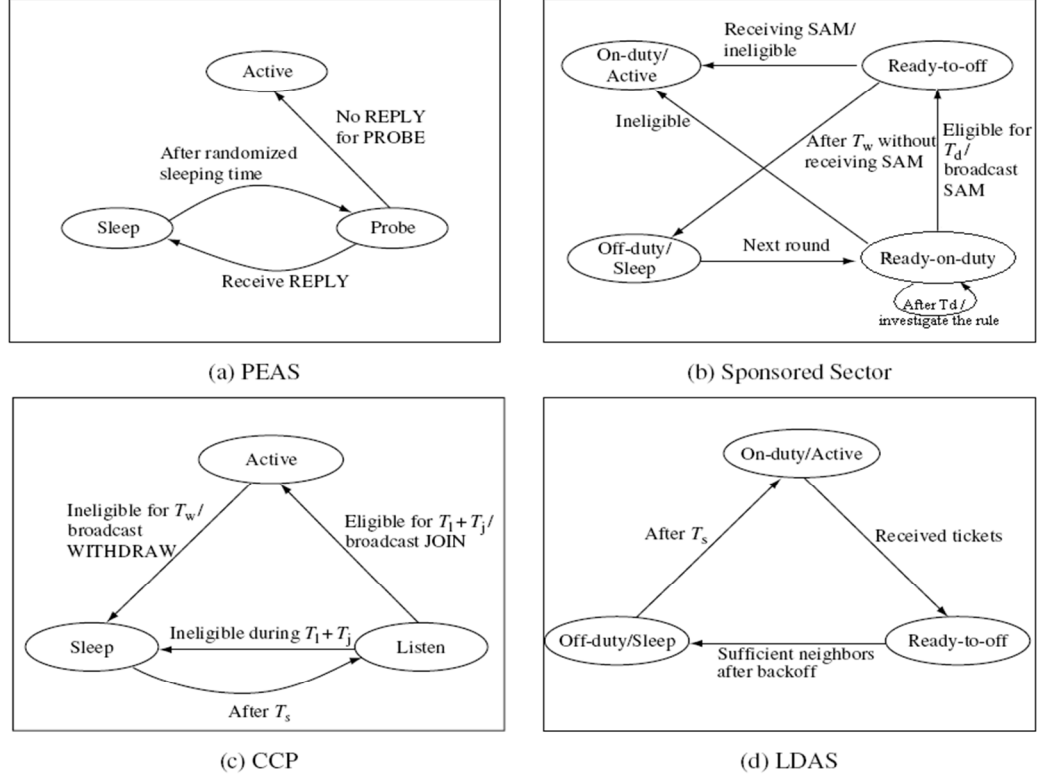


Figure 2.3: State diagrams for sleep-based coverage and connectivity [16]

The sponsored sector [52] approach puts off any redundant nodes while maintaining the original coverage. As explained in Figure 2.3(b), it uses four states: off-duty, ready-on-duty, on-duty, and ready-to-off. It also uses status advertisement message (SAM) that contains the current status of a node, and position advertisement message (PAM) that contains position and sensing range of a node. A node in ready-on-duty state is eligibility to turn-off if it finds from PAM messages that its coverage area is already covered by its neighbors. An eligible node waits for a random backoff time, broadcasts a

SAM message, goes to ready-to-off state, and waits for a timer T_w . If a SAM message is received within T_w , it becomes ineligible and goes to on-duty state; otherwise it can go to off-duty state by turning off its radio and sensor.

The integrated Coverage and Connectivity Protocol (CCP) [39], as shown in Figure 2.3(c), uses three states: sleep, listen, and active. It uses HELLO, WITHDRAW, and JOIN messages; and also few timers. Sensors in sleep state, after a timer T_s , go to the listen state and remain in that state for T_l . They assess their eligibility (to become active) on receiving HELLO, WITHDRAW, or JOIN message. A node is eligible if its region is K -covered. Ineligible nodes go to sleep state after T_l , whereas the eligible nodes start a join timer T_j , and wait for any JOIN message. It goes to sleep state if any JOIN message is received, or otherwise transmit JOIN message and go to active state. All active nodes periodically transmit HELLO messages to tell their active state. Any active node that receives this HELLO message checks its eligibility (to remain active). If it is ineligible to remain active, it waits for withdraw timer T_w , transmits WITHDRAW message, and goes to sleep state. K -coverage implies K -connectivity if $R_c \geq 2R_s$ [39], otherwise a hybrid technique merging CCP with SPAN can be used.

The Lightweight Deployment-Aware Scheduling (LDAS) protocol [53] uses random-voting technique to ensure probabilistic coverage without requiring exact location of neighbors. As shown in Figure 2.3(d), LDAS uses three states: on-duty (active), ready-to-off, and off-duty. A node in on-duty/active state sees the number of working neighbors called n . If n is greater than the required threshold r , then it sends out

penalty tickets to $n-r$ on-duty/active neighbors at random. A node goes to ready-to-off state if it gets number of tickets greater than threshold b , where $b \approx (n-r) - \sqrt{2(n-r) \ln \frac{n+1}{r}}$. After waiting for a random backoff time W_{\max} , it goes to off-duty/sleep state if it still has enough neighbors active. The LDAS algorithm keeps at least r neighbors active.

It is important to note that sleep-based topology control may co-exist with sleep-based MAC protocols, i.e. it is a cross-layer issue. Specifically, MAC-level sleep scheduling should be used only with active nodes, and routing protocols should consider only active nodes.

2.4 Communication Protocols

Communication protocols in WSNs can be made robust and energy efficient by considering state information, such as link quality, link distance, residual energy, location information, and mobility information. This section briefly describes some important routing protocols.

2.4.1 Metric-based approaches

Metric-based approaches ensure robustness by picking routes that minimizes number of transmissions due to failures; and will save on energy.

The *ETX metric* (Expected Transmission metric) and *minimum transmission metric* [54] minimize the expected number of total transmissions on a routing path. ETX

metric involves each node to monitor quality of the links to their neighbors so as to determine routes with minimum expected number of retransmissions due to ARQ. The expected transmission required for successful delivery of a packet on a link can be calculated as $EXT = 1/d_f d_r$, where d_f is the probability of successful delivery and d_r is the probability of successful ACK received. The ETX metric is efficient in terms of both bandwidth and energy.

The *Minimum Outage Route / Minimum Energy Route (MOR/MER)* [55] was analytically developed for highly mobile objects in WSNs. The path selected through MOR is the most reliable, and the path selected through MER is the most energy-efficient. The main benefit of MOR/MER technique is that it does not require nodes to monitor quality of the links.

2.4.2 Routing with diversity

In WSNs, a node broadcasts to possibly several neighboring nodes providing diversity. Relay diversity [55] takes the advantage of wireless broadcast, and improves the reliability by accepting the same packet from two different neighbors. However, this technique has some energy penalty due to overhearing by many receivers.

The *Extremely Opportunistic Routing (ExOR)* [56], also takes the advantage of wireless broadcast, ensures that the node nearest to the destination will forward the received packet. The ExOR has three components: priority ordering of candidate receivers, transmission acknowledgements for finding highest-priority successful

recipient, and forwarding decision in favor of highest-priority node. It minimizes the end-to-end delay and total number of transmissions, has some energy penalty due to overhearing by many receivers.

2.4.3 Multi-path routing

Multi-path routing protocols use multiple disjoint or partially disjoint routes to provide reliable data delivery, and hence save energy.

The “*braided multi-path*” schemes [57] is used for alternate path routing in WSNs when a primary path fails. A braided path does not require node disjointedness between the alternate paths.

In *Gradient Routing Technique (GRAd)* [58], all the nodes in WSN maintain an estimated cost (e.g. hop-counts) to sink. A node will forward the received packets only if its own cost is lesser than remaining cost of the packet. GRAd allows several nodes to forward the same packet, i.e. it is a limited directed flooding. It provides reliability/robustness, but at the cost of greater overhead.

The *GRAdient Broadcast (GRAB)* [59] improves the GRAd technique by assigning credits to hops. It also maintains a cost field through all nodes in the sensor network. Earlier hops are allocated larger share of credits whereas later hops (i.e. closer to sink) are allocated smaller share of the credits. This spreads out path in the beginning but converges at the sink. Each packet has three fields in GRAB: R_o – credit at originating node, C_o – cost to sink at originating node, and U – consumed budget till now.

To prevent routing loops, only receivers with lower costs may forward a packet. GRAB also increase the robustness at the cost of more energy.

2.4.4 Energy-aware routing

Energy-aware routing protocols use residual lifetime of intermediate nodes to select energy efficient routing path. They can be further enhanced by probabilistic forwarding, and flow-based optimizations.

The basic *power-aware routing* scheme [60] chooses nodes with higher reaming battery lifetime as intermediate nodes on the routing path. It finds shortest-cost route by Dijkstra's or Bellman-Ford algorithm using reaming battery lifetime of nodes as cost.

A similar technique is the *lifetime-maximizing routing* [61] that not only selects the minimum energy path, but also avoids the nodes with low residual energy.

The *load-balanced energy-aware routing* [62] uses cost-based probabilistic forwarding, i.e. nodes forward packets only to randomly selected neighbors that are nearer to the sink. This technique provides load balancing by avoiding selection of same nodes every time.

Flow optimization formulations [63] are used to analyze lifetime-maximizing flows and design distributed algorithms for energy-efficient routing. Linear program (LP) is used for this purpose.

2.4.5 Geographic routing

The location information is usually available to sensors through GPS or some localization algorithm. This location-information can be used by geographic routing that is a low-overhead stateless routing.

Local position-based forwarding [64], in its simplest form, forwards the packet to a neighbor located closest to the sink/destination. There are many variants of this scheme, such as Most Forward within R (MFR) [65] that forwards to a farthest neighbor closest to sink, forwarding to the nearest neighbor with forward progress (NFP) [66], and random-forwarding technique that randomly forwards to a neighbor closest to the destination/sink. But all of these techniques face problem when there is no neighbor closer to destination because of the network-hole etc. For this purpose, there is *greedy-face-greedy (GFG) algorithm* [67] that is the basis of the greedy perimeter stateless routing protocol (GPSR) [68]. GFG routes packets along the face of a planar sub-graph using the right-hand rule. A packet shifts from greedy to face-routing mode whenever it arrives at a dead-end, and then returns back to greedy mode when the dead-end finishes.

The *PRR*D metric* [69] dictates nodes to forward packets to a neighbor with highest PRR*D metric, where PRR is the packet reception rate, and D is the distance improvement towards the destination.

The *Geographical & Energy-Aware Routing (GEAR)* [70] was designed to forward queries from a sink to all sensor nodes in a specific region. Another similar

technique is the *trajectory-based forwarding (TBF)* [71] used for routing of queries.

2.4.6 Cluster-based routing

One of the well-known cluster-based routing protocols [72], [73] aims to mimic the cellular network. Gateways are used to maintain communication between two or more clusterheads. Selecting the appropriate gateway is an open point for discussion. In current cluster-based routing protocols, selecting gateways is mostly random, which is not a power aware approach. It has been documented that using small hops is optimal to reduce the transmit power level [74], [75], if one wants to increase the traffic carrying capacity of the entire network. Hop count was typically the measure of routing; therefore, a constant metric per hop was used. The constant metric should be replaced with a power metric where mobile terminals adjust their transmit power depending on the distance of their neighbors [76], [77].

The Common Power (COMPOW) protocol [78], [79] proposes that all the nodes transmit at the same power to ensure bidirectional links. The transmit power is the minimum power level that ensures connectivity of the network. COMPOW shows that choosing the smallest range, subject to maintaining network connectivity, maximizes the traffic carrying capacity. However, power control also impacts on battery life time. COMPOW works as follows. First, multi-hop shortest path algorithm (such as Bellman-Ford or Dijkstra algorithms) is performed, one at each possible power. Each node picks the lowest power level such that the number of reachable nodes is the same as the number of nodes reachable with the maximum power level. After a few iterations, the network

will converge to the lowest power that ensures connectivity of the network. One major drawback is by enforcing common powers; one isolated node will cause all the other nodes to have excessive large power level.

Another protocol described in [80], which is a Cluster Power (CLUSTERPOW) protocol that divides the network into virtual clusters with no clusterheads. It is proposed to overcome the drawbacks in COMPOW by allowing different clusters to have several power levels. The COMPOW protocol can be considered as a special case of CLUSTERPOW protocol. The network consists of hops of different transmit power such that it is divided into clusters according to transmit power. However, the performance of the CLUSTERPOW protocol was not thoroughly tested. The only application in [80] was with Destination Sequenced Distance Vector (DSDV) protocol, which is proven to be very inefficient in ad hoc networks. Moreover, the protocol itself is rather complex. A simulation experiment in [78] shows that low power transmission does conform with power aware routing, since power aware routing prefers many short hops to a long ones. Also, another simulation experiment in [78] shows that low power level minimizes the contention on the MAC layer.

The Warning Energy Aware Clusterhead (WEAC) protocol [73], divides the network into clusters. Unlike the CLUSTERPOW, each cluster is managed by a clusterhead; nonetheless, neighboring nodes can communicate with each other without consulting their clusterheads. Communication between clusters is maintained by gateways.

In the above protocols, sleep scheduling techniques have not been addressed, which means that all nodes have to be active all times; thus, wasting unnecessarily power.

In the Neighbor-Aware Clusterhead (NAC) protocol [81], nodes are scheduled for sleep periods without neither affecting the connectivity nor the coverage of the network. Clusterheads manage the synchronization and sleep scheduling issues within their clusters. Gateway may choose to have more frequent wakeup times to satisfy the connectivity and coverage issues. A clusterhead may act as gateway if it is communicating to two different zones. Cluster-based protocols organize the network into a hierarchical structure to manage the network in an efficient way.

2.4.7 Routing to mobile sinks

In case of moving sink, different protocols for forwarding are needed. One can benefit from research in mobile ad hoc networks (MANET) where many protocols, such as DSR, AODV, ZRP, ABR, and TORA have been developed to provide robustness in the presence of mobile nodes or dynamic topology. A mobile sink can be very useful in gathering of timely information from remotely deployed sensors, and may also result in energy conservation.

The *Two-Tier Data Dissemination (TTDD)* [47] makes a grid overlay dissemination network on top of static network, where grid points act as data dissemination nodes. A source sends data to sink using this overlay network to its nearest data dissemination node that forwards it to primary agent of the mobile sink.

The *Scalable Energy-efficient Asynchronous Dissemination (SEAD)* [82] can forward data to several mobile sinks in a WSN. Each mobile sink declares nearby static node as access node (AN) for communication. This access node is responsible for tracking the mobile node in nearby area; otherwise a new access node is declared by the mobile sink.

The *Mobile Ubiquitous LAN Extensions (MULE)* [83] provides connectivity using mobile nodes that may act as mobile sinks or help in data transfer between sensors and any fixed access points. The MULE nodes may randomly walk in the network. Nodes in the network transfer all data to MULE nodes on coming closer to each other, whereas the MULE nodes transfer all data to access point on contact. The delivery rate can be increased by using more MULE nodes. However, using fewer MULE nodes with increased buffer size may compensate.

The *Hybrid Learning Enforced Time Domain Routing (HLETDR)* [84] technique is useful when mobile sink takes fairly predictable path but follows a stochastic repeated pattern. Nodes near that path are called *moles* which can learn the probability distribution of the sink's appearance in their area. Data is forwarded to a mole that is more likely to encounter the mobile sink.

CHAPTER 3.

SAND AND DUST STORM

DETECTION

Sand and dust storms (SDSs) are common phenomena in many parts of the world, which have terrible effects on our daily life. Several technologies are available for monitoring of SDSs, such as lookout-tower, video-surveillance, sensory information, satellite imagery, unmanned aerial vehicle and hybrid approaches. This chapter provides a quick review of techniques and technologies for sand and dust storm monitoring.

3.1 Introduction

Sand and dust storms (SDSs) are common phenomena in the Middle-East, North-Africa, North-China and Australia during spring, winter and early summer [85]. They disrupt our daily life, and put our environment, economy and health at risk. By injecting dust into the air, sandstorms cause pollution, respiratory diseases, ecological disaster, low-visibility and interrupted transportation and aviation. As the sandstorms accompany strong wind, they cause desertification by wearing away the soil-surface, and cause huge economic loss by damaging the structures. Finally, by depositing dust on the ground, they reduce grain production by affecting the photosynthesis, and spread viral diseases by

carrying virus-spores from source to many other places [86–89]. An accurate and early warning of the forthcoming SDS would minimize its severe consequences by giving people sufficient time for taking any precautionary actions, such as seeking shelter, harvesting ripen crops, etc.

The main sources of dust are deserts, arid or dry regions, terrains with loose dirt and uncultivated lands [90]. Dust storms are created when strong surface wind uplifts fine-grained dust particles into the air, strong turbulence or convection diffuses this dust to a large area, and strong wind carries this dust over long distances [91]. Similarly, sandstorms drift huge amount of larger sand particles nearer to the surface. Although, there is a slight difference between sandstorm and dust storm, these two terms are used interchangeably to represent any or both of them.

There are many types of SDSs with respect to spatial and temporal coverage. A local or small-scale SDS emerges in a small geographic area, while a global or large-scale SDS covers a very large geographic area. A short-term SDS is the one that persists for a few minutes or hours, whereas a long-term SDS would prevail for many days or months [92]. However, based on the intensity of SDS, World Meteorological Organization (WMO) has classified dust events into four types: Dust-in-Suspension (visibility: up to 10 km), Blowing Dust (visibility: 1 km to 10 km), Dust Storm (visibility: 200 m to 1000 m) and Severe Dust Storm (visibility: less than 200 m) [93]. The physical parameters of SDS include its optical depth, concentration, particle size distribution (PSD) and land surface cover below it [94].

The main issues in developing a system for SDS detection and prediction include defining data requirements, modeling dust and designing an effective prediction technique. Such systems require data on dust and other environmental changes, which can be obtained in two ways: ground-based observations and spaceborne observations [95]. The ground-based observations can be made through lookout towers, video surveillance and sensory information gathered by radars, lidars, WSNs, etc. The spaceborne observations are typically obtained through satellite imaging, and sometimes through unmanned aerial vehicle (UAV), etc. However, hybrid approaches make use of the both types of observations for better performance. Figure 3.1 classifies the most commonly used technologies for SDS detection and prediction.

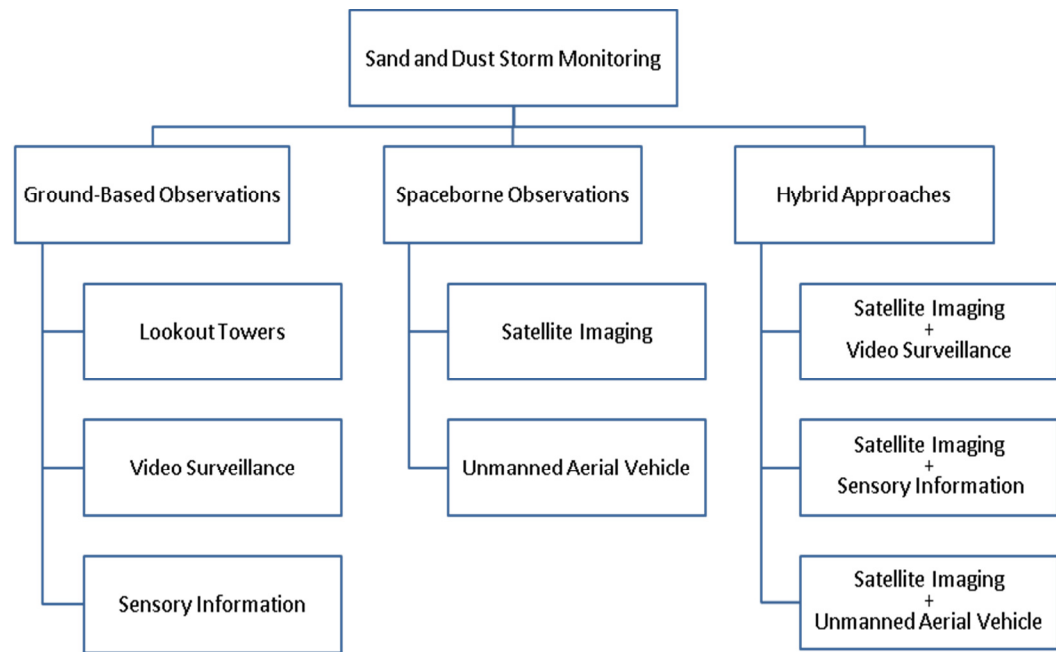


Figure 3.1: Technologies for Sand and Dust Storm Monitoring

To the best of our knowledge, this is the first study that provides a brief but comprehensive review of the techniques and technologies for SDS detection and prediction.

The rest of this chapter is organized as follows. Section 3.2 explains some basics of detection and prediction of SDS. Section 3.3 and 3.4 describe SDS detection methods based on ground-based observations and spaceborne observations respectively, while hybrid approaches are discussed in section 3.5. A comparison of different technologies is provided in section 3.6. The discussion is concluded in section 3.7 at the end.

3.2 Basics of SDS Detection

The systems for SDS detection and prediction need to continuously monitor the fugitive dust, sandstorms, movements of sandbanks and other environmental changes in a certain area. Several technologies are available for dust monitoring, such as satellite remote sensing, video surveillance, wireless sensors, etc. A particular technology is suitable only for detecting certain types of SDS with respect to spatial and temporal coverage. For example, satellite methods are very effective in detecting the large-scale and long-term SDS, but inefficient in detecting the small-scale and short-term SDS due to poor spatial and temporal resolution of the images [95]. After selecting a suitable technology for dust monitoring, the main issues in developing a system for SDS detection and prediction include data requirements, dust modeling and prediction technique.

3.2.1 Data Requirements

The data requirements for SDS detection systems vary from one technology to another. For example, SDS detection using lookout-tower requires human observations on regular basis; UAV and video-surveillance methods need visible images captured by digital cameras; satellite-based methods require multi-spectral images captured by satellite sensors; sensory-information methods require several kinds of field data from radars, lidars or sensor networks; and hybrid methods require data from several technologies in use.

The SDS can be easily detected from digital images by applying simple techniques, such as digital image processing (DIP), artificial intelligence (AI), etc. The multi-spectral images from satellites use several bands such as optical/visible or red-green-blue (RGB), near-infrared (NIR), middle infrared (MIR) and far infrared (FIR) or thermal, and microwave or radar [96]. Therefore, the multi-spectral images from satellite sensors may require additional techniques based on, for example, optical depth, brightness temperature, thermal properties, refractive index, etc. [97], [98].

Radar and lidar actively transmit radio signal in the microwave band and laser light in the visible band respectively, and record how much of it is reflected back. The radar/lidar data can be processed, which consists of many columns including X- and Y-coordinates and Z-elevation, to create an image of the environment in terms of surface elevation. While the spaceborne radars/lidars are used to produce the land surface point

elevation, the ground-based radars/lidars are commonly used for mapping of clouds, dusts, etc. [95], [99–101].

For getting accurate, first-hand, up-to-date and continuous observations from the region of dust-origin, the use of wireless sensor networks (WSNs) is becoming very common nowadays. The data required by a WSN-based SDS detection system includes [102]:

- 1) ***Atmospheric pressure:*** It is the measure of pressure at any point exerted by the atmosphere of earth. It plays an important role in creating storms, developing clouds, and blowing wind by moving the air from high pressure areas (with low temperature) to low pressure areas (with high temperature).
- 2) ***Surface temperature:*** It is the measure of air temperature at 2 meters above the ground-level. It shows how much the surface is dry and hot, and it affects the amount of dust the wind may lift.
- 3) ***Humidity:*** It is the amount of moisture present in the air at 2 meters above the ground-level. This moisture may capture some of the blowing dust to make aerosol dust that can travel several miles with the wind.
- 4) ***Wind velocity:*** It is the speed and direction with which air moves from high pressure areas to low pressure areas. A strong wind can lift the dust from a surface and transport it to other remote areas.
- 5) ***Soil moisture:*** It is the measure of water contents in the upper part of soil layer. It affects the growth of vegetation, soil erosion, etc.

Additionally, WSNs provide other useful information includes lighting or visibility, time and location. It is important to note that strong wind velocity is a necessary, but not sufficient, condition for the creation of dust storm [91], [103].

3.2.2 Dust Modeling

A dust prediction model uses numerous characteristics of land, dust and environment in any geographic region of interest. The nature, reasons, and characteristics of SDS have been studied extensively since 1980's [104–107] resulting in several dust prediction models, which reflect the characteristics of an actual dust storm. There are many types of dust models, which include global dust models [108–112], regional dust models [113–120], meso-scale dust model [121], [122], and integrated dust models [123–125]. However, the modern research in this field is directed towards the development of integrated dust models which incorporate all the related models, including the following [123]:

- 1) ***Atmospheric Model***: It includes atmospheric dynamics and physical processes, such as radiation, clouds, convection, diffusion, etc., and it also affects all the other models.
- 2) ***Land Surface Model***: It includes data on soil particle-size distribution, soil surface characteristics such as moisture and temperature, vegetation coverage, leaf-area index, and energy and mass fluxes.
- 3) ***Dust Emission Model***: It includes threshold velocity of wind, saltation flux and dust emission rate.

- 4) **Dust Transport:** It includes data on horizontal and vertical advection, vertical diffusion, and dry and wet convection.
- 5) **Dust Deposition:** It includes data on dry and wet deposition of the dust.
- 6) **External Components:** The other useful components of an integrated dust storm model are Geographic Information System (GIS) databases, dust monitoring technologies, etc.

The most important factors in these dust models are: threshold velocity of wind, particle-size distribution (PSD), dust emission rate and surface conditions [126]. The threshold velocity of wind can be as small as 4 m/s in Tarim Basin of China [123], or as large as 8 m/s for some parts of the United States [127]. Although its value depends on the region, it is assumed to be 6.5 m/s in several dust models [90]. However, for strong and large-scale SDS, the value of threshold velocity of wind is usually between 10 to 14 m/s depending on geographic region [90], [127], [128]. The dust emission rate is determined by particle-size distribution, wind speed, saltation flux, surface roughness, etc. [112], [129], [130]. The particle size distribution (PSD) shows the distribution of dust particles in a dust storm. The particle size of airborne dust is much smaller than desert sand and dust, i.e. less than 5 μm versus greater than 70 μm respectively [131]. The surface conditions, such as vegetation cover, surface crusts, soil moisture, soil temperature, etc., vary greatly from region to region such as cultivated, semi-arid and dry arid [90].

Statistical techniques such as regression, maximum likelihood, Bayesian networks, support vector machines, etc., and artificial intelligence techniques such as neural networks, fuzzy logic, genetic algorithms, pattern matching, etc., are commonly used for dust modeling, classification and prediction. For example, a sandstorm forecasting model using artificial neural network (ANN) and genetic algorithm (GA) has been developed, and proved as stable, accurate and fast [132]; and a dust prediction model based on lognormal particle size distribution (PSD) has been proposed to reproduce the actual variation of the dust particle size in different dust models and to predict the signal attenuation during dust storms [133].

The dust prediction modeling is very useful in many SDS applications such as prediction of dust storms, study of dust processes, measurement of the global dust cycle, remodeling of the past climate, etc. [123].

3.2.3 Prediction and Early Warning

To help minimize the hazards of SDS, it is very important to forecast the SDS and then issue an early warning to the public allowing them to take preventive measures. Some studies focus on the prediction of short-term SDS while others focus on long-term SDS [92].

Some dust prediction and early warning systems use indexes to show the size, frequency and intensity of dust phenomena, such as DSI – Dust Storm Index [90] which is calculated as $DSI = (5 * SD) + MD + (LDE / 20)$, where LDE is the frequency of local dust

events, MD is the frequency of moderate dust storms and SD is the frequency of severe dust storms. Similarly, FDE – frequency of dust events [134] analyzes dust events in Mongolia and is calculated as $f_{DE}=f_{DIS}+f_{BD}+f_{DS}+f_{SDS}$, which is the sum of all dust events for Dust-in-Suspension, Blowing Dust, Dust Storm and Severe Dust Storm [93]. Other examples of such indexes include IDDI – infrared difference dust index [135] which uses Meteosat-IR imager to exploit the impact of dust aerosols on the thermal infrared radiance, and NDDI – normalized difference dust index [136] which uses reflectance measurements by MODIS to detect dust over bright surfaces and distinguish dust from clouds.

Several methods are available for the prediction of short-term SDS, which include numerical forecast models [125], [137–140], analysis of satellite images [141], synoptic system analysis [142], discrimination function analysis [143], [144], genetic algorithms [145] and artificial neural networks [146], [147], etc. Similarly, several methods are available for the prediction of long-term SDS, which include trend-based forecasts [92], [148], numerical forecast models [149–151], environmental modeling [137], [138], [152–154], integrated dust storm prediction systems [123–125], [128], [155], etc.

For an early warning of upcoming SDS, the integrated dust storm model would also utilize GIS databases along with dust monitoring and other technologies. For example, an integrated dust storm model [123] has been developed based on numerical models, satellite remote sensing, synoptic-observations and GIS data. Similarly, NASA has used data from Aqua and Terra satellites to improve an existing dust model regarding

the estimation of wind velocity, surface temperature, and the origin and amount of dust flown, in order to improve SDS prediction under the Public Health Applications in Remote Sensing (PHAiRS) project [155].

3.3 Climate Implications of SDS

Sand and dust storms severely pollute our environment by suspending tiny sand and dust particles, called as dust aerosols, in the atmosphere. Several satellite based instruments are available to measure the optical thickness of dust aerosols, called as aerosol optical depth (AOD), to identify the presence and intensity of dust in the environment. Figure 3.2 shows the global map of AOD at 550 nm on April 6, 2012. Dust can be noticed over China, Middle-East, and North Africa.

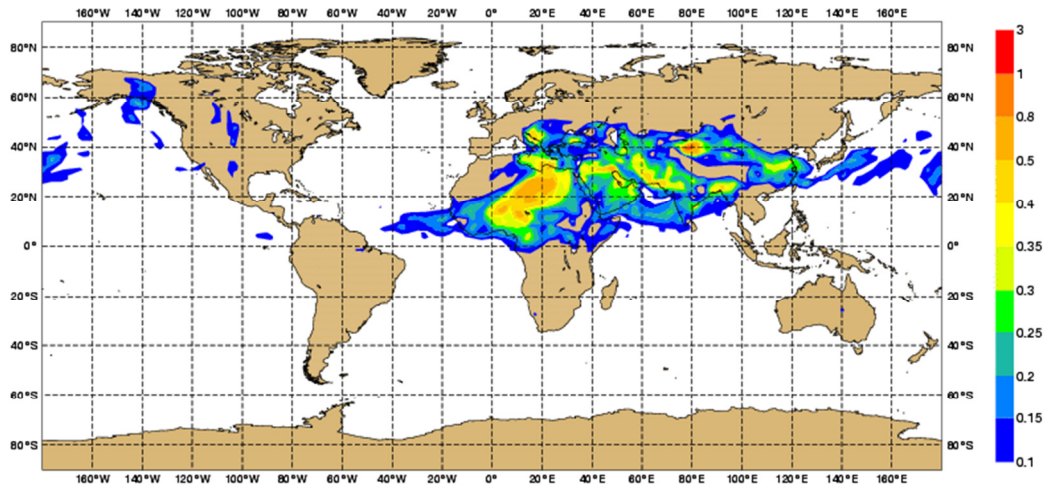


Figure 3.2: Global Map of AOD at 550 nm for April 06, 2012 00UTC

[Source: <http://www.gmes-atmosphere.eu>]

The dust aerosols can have very important forcing effects on radiative budget of the earth by exerting a radiative influence on climate in two ways:

1. Absorbing and reflecting the sunlight: Dust aerosols cause cooling by reflecting the incoming sunlight back into space or by absorbing it.
2. Acting as condensation nuclei: Dust aerosols act as condensation nuclei and help in formation of ice and clouds. The clouds in turn keep the long wave radiations from the earth within the atmosphere.

SDSs have local and global effects on climate changes, while climatic changes in turn can change the position and intensity of dust sources in the world. Table 3.1 shows global statistics on SDS based on [156]. There exists an inverse relationship between aerosol index and annual average rainfall, while a direct relationship between aerosol index and dust emission.

The main source of global dust is Sahara Desert with 69% of the total dust emission worldwide, while Europe and Antarctica contribute almost nothing to mention. As SDSs can transport to remote areas, it is very difficult to find the frequency of SDS in each region. However, the frequency of SDS can be described for each type of land-surface or vegetation [157], as shown in Table 3.2. It is important to note that regions with bare grounds and rough surfaces have highest frequency of SDSs, while regions covered by vegetation have the lowest frequency.

TABLE 3.1: Region-wise sand and dust storm statistics

Region	Major source (s) of dust	TOMS-AI (Aerosol Index)	Annual Average Rainfall (mm)	Dust emissions	
				Tg/year	%
North Africa	Sahara Desert	>1.5–3.0	17–100	1430	69.0
Asia	Makran coast of Pakistan, Taklamakan Desert or Tarim Basin of China, Arabian Desert.	>1.1–2.1	25–100	496	23.9
Australia	Lake Eyre Basin	>1.1	150–200	61	2.9
South Africa	Etosha Pan of Namibia, Mkgadikgadi Basin of Botswana.	>0.8–1.1	460–530	322	1.1
South America	Salar de Uyuni of Bolivia	>0.7	178	55	2.7
North America	Great Basin of the USA	>0.5	400	9	0.4

TABLE 3.2: Frequency of SDS worldwide for each type of land-surface

Vegetation or land-surface type	Frequency of SDSs (days/year)
Bare ground	6.8
Shrubs and bare ground	2.6
Grassland	1.6
Cultivated crops	0.8
Broadleaf deciduous forest and woodland	0.3
Wooded grassland	0.1
Forest	0.1

The next two sections provide a detailed study of the common technologies for SDS detection along with techniques. The pros and cons of each technology are also discussed inline.

3.4 Ground-Based Observations

The traditional methods for ground-based observations include lookout towers, video surveillance and sensory information. The ground-based observations are more accurate and detailed. However they are mainly limited to the land, and are unable to identify the transport of dust on large-scale.

3.4.1 Lookout Towers

The lookout tower, also known as observation tower, is perhaps one of the oldest methods used for weather monitoring. A lookout tower is a high structure used to observe the weather or other events, such as fire, dust, etc., from a long distance. They allow a human observer to have a 360° view of the surrounding areas [158].

The ability and accuracy of detecting SDS using a lookout tower is extremely limited by human capabilities. It is not feasible to establish and manage thousands of such towers in remote areas. Most of the existing towers are either abandoned or being used as museums, wildlife lookouts, tourism places nowadays.

3.4.2 Video Surveillance

Digital images from a ground-based camera station are useful in detecting SDS in its surrounding areas. A camera-station is equipped with video cameras installed locally or remotely. The images received from these local or remote cameras are processed for detecting dust events, and an alarm is issued in case of any approaching SDS. Simple DIP and AI techniques can be used for SDS detection from digital images.

The video surveillance technology can be used to detect any kind of dust events except large-scale SDS. However, it can also be used to capture the startup phase of large-scale SDS [159]. It works even during the cloudy days when satellite imaging cannot do well. The camera stations are usually used to augment remote sensing satellites in order to enhance their capability of detecting SDS [127].

3.4.3 Sensory Information

Radar, lidar and WSNs are the most commonly used sensors for obtaining information about the environment. While WSNs are typically used for ground-based observations, radar and lidar are used for both ground-based observations in weather stations and spaceborne observations in UAVs and satellites.

3.4.3.1 Radar and Lidar

A RADAR (Radio Detection And Ranging) actively transmits radio signal in the microwave wavelength (300 MHz to 30 GHz) and records how much of it is reflected back. Through computer processing of the reflected signal using Doppler Shift, it can find the height, distance, size, speed and direction of objects, and can create a visible image of the environment. It works fine both at daytime and nighttime, and also in bad weather conditions because radar signals can penetrate clouds, fog, dust, etc.

Similarly, LIDAR (Light Detection And Ranging) actively transmits either infrared or laser light in the visible wavelength and then records how much of it is reflected back. It uses time of flight (ToF) and/or Doppler Shift to create an image of the environment and to find the height, distance, size, speed and direction of objects. However, it doesn't work well in bad weather conditions due to use of light.

Both radar and lidar have been used extensively for dust detection. For example, weather radar observations and diagnostic analysis have been used to detect SDSs in autumn [100]; ground-based lidar systems along with satellite imaging, regional dust

models and particle samplers have been used to monitor a severe dust storm over Greece [101]; spaceborne lidar measurements have been used to detect dust aerosols [95]; and lidar data over Western Europe have been used for dust detection [160].

3.4.3.2 Wireless Sensor Network (WSN)

Previously, WSNs have been extensively used for monitoring of natural hazards such as volcanic eruption, earthquake, flood, snow, forest and other fires, etc. [161–166]. However, the idea of using WSNs for SDS detection is very new.

There are many advantages of using WSNs for dust monitoring: 1) sensors can be deployed in areas where satellite signals are not available, 2) the same network can be used for detecting other events such as fire, rain, snow, volcanic eruption, etc., 3) they provide more accurate, up-to-date and detailed data, and 4) they provide quick information about the origin of dust. However, many serious and unique issues arise in using WSNs for SDS detection, such as:

1. ***High signal attenuation during SDS:*** The higher signal attenuation during SDS may disturb the communication in WSNs. The sensor nodes in WSNs use radio signals for communication, which may face undesirably high attenuation during SDS. However, prediction models for microwave attenuation can be very useful in such situations, for example, the mathematical model for prediction of attenuation in microwave signal during SDS [167]. Similarly, signal attenuation has been predicted by using the lognormal particle size distribution (PSD), instead of assuming a fixed size of dust particles, while considering the variation of particle size of sand or dust inside SDS [133].

2. *Sensor nodes buried under sand during SDS:* Many of the sensor nodes on ground may be buried under sand during SDS event. These buried sensors may significantly degrade the connectivity of network. In order to enhance the network connectivity during sandstorms, it has been proposed to use multiple available channels including air-to-air (AA) channel, air-to-sand (AS) channel, sand-to-air (SA) channel, and sand-to-sand (SS) channel [102].

These recent developments have made it feasible to use WSNs in SDS monitoring systems. Accordingly, modern SDS monitoring systems use WSNs as an important component along with spaceborne observations using satellites.

3.5 Spaceborne Observations

The spaceborne observations provide very wide or global coverage of dust events and transfers, but have low performance at night and in cloudy environments. The most commonly used methods for spaceborne observations include satellite imaging and unmanned aerial vehicles (UAVs). TABLE 3.3 provides a comparison of the capabilities of some popular satellite imaging devices.

3.5.1 Satellite Imaging

Satellite imaging or remote sensing has become one of the most commonly used approaches to detect large-scale and long-term SDS. Several satellites make their data freely available to everyone, through either real-time broadcast for regional coverage or the Internet for global coverage. Unfortunately, the resolution of this data is lower than

what most scientific applications need. For example, MODIS provides data of resolution ranging from 250 to 1000 meter/pixel while GOES and AVHRR both provide data of 1000 meter/pixel or worse. The large-scale and/or long-term SDS can be detected by using images from several devices or sensors onboard different satellites. A comparison of the capabilities of a few satellite imaging devices is provided in Table 3. The most commonly used satellite imaging devices for SDS detection are MODIS, AVHRR, GOES and SEVIRI.

MODIS – MODerate resolution Imaging Spectroradiometer – [168] is an imaging instrument that is carried by both Aqua and Terra satellites of NASA. It allows continuous monitoring of the environment by providing information on several parameters using 36 spectral bands, including aerosol density, atmospheric trace gases, and mapping of clouds, ocean and land. There are more than 50 weather stations worldwide which receive MODIS broadcasted data using special dish antennas. MODIS imaging data is also available through their website. AVHRR – Advanced Very High Resolution Radiometer – [169] is a sensing device onboard the series of NOAA-POES satellites (National Oceanic and Atmospheric Administration - Polar-orbiting Operational Environmental Satellites), including NOAA-1 through NOAA-17. They are the meteorological backbone of United States. Currently, a total of nine satellites, NOAA-9 through NOAA-17, are operational and use four (i.e. AVHRR/1), five (i.e. AVHRR/2) or six (i.e. AVHRR/3) spectral channels, while all the later satellites will have six channels. AVHRR/3 provides information on day/night mapping of clouds and surface, detection of ice/snow and land-water boundaries, and temperature of sea surface. GOES –

Geostationary Operational Environmental Satellites – [170] are geostationary satellites that are high enough to provide full-disc view of the Earth. There are fifteen satellites in this series, GOES-1 to GOES-15, where only four satellites, GOES-11 to GOES-14, are working now and GOES-15 is in testing mode. GOES satellites can watch the developments of large-scale storms and follow their movements across the globe. SEVIRI – Spinning Enhanced Visible and InfraRed Imager – [171] is a Meteosat Second Generation (MSG) instrument which can observe the Earth in 12 spectral bands. Eight of the bands are thermal infrared (TIR) which provide more precise data about the temperatures of clouds, land, sea surfaces, etc. Meteosat MSG is an European geostationary satellite that has temporal resolution of 15 minutes and provides some unique services. For example, Meteosat RGB Composites Dust images can clearly identify SDS with higher red fraction in the day/night images.

The main problem with the commonly used satellite sensors is that they produce data having very low spatial and temporal resolution. This issue is taken by the sensorweb project [172], which uses low-resolution devices to track large-scale SDS and separately triggers high-resolution devices, such as EO-1, to capture more detailed data. Similarly, to improve SDS prediction using satellites, NASA's Public Health Applications in Remote Sensing (PHAiRS) project [155] uses measurements from sensors on Aqua and Terra satellites to improve an existing dust model in estimation of the wind velocity, surface temperature, and the origin and amount of dust flown.

TABLE 3.3: Comparison of the capabilities of some satellite imaging devices

Satellite / Sensor	Owner	Since	Spatial resolution		Temporal resolution		Spectral resolution	
			Meters / pixel	Rating	Orbit/ Interval	Rating	Bands	Rating
GOES	NOAA, USA	1994	1000, 4000, 8000	Poor	15 min	Excellent	5	Good
MODIS	NASA, USA	1999	250, 500, 1000	Fair- Poor	1 day	Fair	36	Excellent
AVHRR	NOAA, USA	1978	1000	Poor	1 day	Fair	5	Good
Meteosat SEVIRI-MSG	EUMETSAT, EU	2004	1000 (HRV), 3000	Poor	15 min	Excellent	12	Excellent
Landsat TM	NASA, USA	1982	30, 120	Excellent	16 days	Poor	7	Good
WiFS	ISRO, India	1995	23, 70, 188	Excellent	24 days	Poor	4	Fair
SeaWiFS	OrbImage, USA	1997	1130	Poor	1 day	Fair	8	Good

SPOT	Spot Image, France	1986	5, 10, 20	Excellent	26 days	Poor	3	Fair
RADARSAT	CSA, Canada	1995	8-100	Excellent	24 days	Poor	1	N/A
Earlybird	DigitalGlobe, USA	1997	3, 15	Excellent	2 days	Fair	3	Fair
IKONOS	GeoEye, USA	1999	1, 4	Excellent	2 days	Fair	4	Fair
ASTER	NASA, USA	1999	15, 30, 90	Excellent	4 days	Fair	14	Excellent
MISR	NASA, USA	1999	275	Fair	16 days	Poor	4	Fair
JERS-1	NASDA, Japan	1992	18	Excellent	44 days	Poor	7	Good
MERIS	NASA, USA	2001	300, 1200	Fair- Poor	35 days	Poor	15	Excellent
Quickbird	DigitalGlobe, USA	2001	0.61, 2.44	Excellent	1 day	Fair	4	Fair

AMSR-E	NASDA, Japan	2002	5400 - 56000	Poor	12 hours	Good	12	Excellent
OrbView-3	OrbImage, USA	2003	1, 4	Excellent	3 days	Fair	4	Fair
GERB	UK, Belgium, and Italy	2002	44.5km×39.3km	Poor	15 min	Excellent	2	Fair
TOMS (Earth Probe)	NASA, USA	1996- 2005	50 km × 50 km	Poor	1 day	Fair	6	Good
OMI (EOS-Aura)	Netherlands	2004	13 km x 24 km, 13 km × 12 km	Poor	1 day	Fair	2	Fair

Qualitative rating: poor, fair, good, excellent.

Application of satellite imaging in SDS detection has been extensively studied in the last two decades. The satellite imagers make images by using reflectance. While optical satellites sense the sunlight reflected by any object, microwave or radar satellites sense the reflections of microwave energy that is transmitted by them towards any object. It is important to note that dust storms have high reflectance and hence appear fairly bright in visible spectrum, but react strongly to scattering and hence have decreased brightness temperature (BT) in microwave spectrum; and that the reflectance of snow cover, cloud and water is higher than that of dust storms and land surfaces in both visible and near-infrared spectra [173]. The radiance temperature or BT is a measure of the intensity of radiation thermally emitted by an object, while a related term, 'optical depth', measures the opacity of an object where any value lesser than 1 means that the object is relatively transparent and any value greater than 1 means that the object is opaque [174]. In the infrared wavelength region, dust shows a decreased BT in all weather conditions. Because BT depends on wavelength and doesn't depend on the amount of dust, it cannot be used for dust detection [175]. However, brightness temperature difference (BTD), a very useful BT-based method, is commonly applied for dust detection which uses the difference in BT of two or more wavelength pairs. Several BTD methods are available for varying purposes. For example, BTD (8.6-11 μm) is used to detect dust and clouds because it is highly sensitive to both dust and clouds, but BTD (11-12 μm) is used to distinguish dust from aerosols and clouds because it has negative values for dust but positive values for aerosols and clouds [176]. Similarly, Meteosat provides RGB Composites which can clearly identify dust storms with higher red fraction in the

day/night images by using the difference of channels as $R=IR12.0-IR10.8$, $G=IR10.8-IR8.7$, and $B=IR10.8$ [171]. Moreover, advances in DIP and AI have also produced many new methods for detection of dust storm, its source, direction, etc., using satellite images [177], [178].

3.5.1.1 Visible and near-infrared (VIR) imaging

Visible and near-infrared (VIR) imaging offers some difficulty in SDS detection over the originating regions with bright land surfaces, restricting their application to over water such as oceans and dark land surfaces such as vegetation areas [179]. To beat this restriction, many approaches have been proposed to find optical properties of aerosol over land surfaces using MODIS, for example [180–182]. Mostly, aerosol applications use dark-target approach to detect brighter aerosol over land surfaces with lower reflectance, which again limits the application of MODIS in dust originating regions [180]. Although BTD can detect pure dust over both sea and land surfaces that solves these issues, it cannot detect dust covered by clouds, and also fails to detect dust over dusty-land and Yellow Sea having golden-yellow surface of water due to the sand deposited by storms from Gobi Desert [183].

Recently, normalized difference dust index (NDDI) [136] has been proposed that can detect dust over bright surfaces and distinguish dust from clouds using reflectance measurements by MODIS. The NDDI has been tested over both Asian and African regions. Another recent method, called as decomposition of unpolarized infrared reflectivity [183], breaks unpolarized reflectivities obtained through MTSAT-1R satellite into vertical and horizontal reflectivities, called as V- and H-polarized reflectivities, in

order to differentiate between dust and cloud over the sea by using H-polarized reflectivity and the simulated sea surface temperature (SST). Similarly, RST – robust satellite technique – [178], [184] analysis the satellite data to detect natural and environmental hazards and has been applied on MTSAT-1R images to detect SDS in Australia, both over land and sea.

It is possible to find the presence, origin, direction and speed of dust storms in satellite images using DIP and AI techniques. For example, PCA-based data fusion – a data fusion technique based on Principal Component Analysis (PCA) – [185] can identify the presence of dust storms in MISR images by combining several images to create a new image of higher quality. The improved shape of the PCA-based data fusion technique is HPCA – Hierarchical PCA – [186] which is computationally much faster. Similarly, the center and intensity of SDS can be found from MERIS imagery using 2D-Otsu algorithm, fuzzy entropy and particle swarm optimization (PSO) algorithm [178]; and the presence of dust storm, its source and direction can be detected in AVHRR images using DIP techniques such as directional analysis, filtering, etc. [177].

3.5.1.2 Thermal infrared (TIR) imaging

Thermal infrared (TIR) channel is very useful, perhaps the best, in detecting SDSs. Dust storms often take place during the nighttime when VIR and NIR images are not much helpful in detecting the dust event. Therefore, thermal imaging is necessarily provided by many satellite sensors. The thermal-based approaches, for example BTM, use thermal properties of mineral aerosols, background and water vapors in order to distinguish between them [180]. However, this technique faces difficulty in

differentiating dust aerosols from dust originating regions such as deserts, semiarid and arid areas because dust aerosols and dust sources have similar brightness temperature.

Fortunately, the particle size of airborne dust is much smaller than that of desert sand and dust, i.e. less than 5 μm versus greater than 70 μm respectively, which results in a strong difference in their infrared emissivity. This difference in particle sizes has been successfully used to detect airborne dust over deserts using MODIS thermal channels, both at daytime and nighttime [131]. Recently, this technique has been used to detect dust events over UAE using imagery by METEOSAT SEVIRI-MSG that is an European geostationary satellite having temporal resolution of 15 minutes, i.e. 96 images per day [187], [188]. A similar approach uses refractive index and particle size of aerosol in order to detect SDS [189].

Both VIR and TIR techniques have been used for SDS detection since 1970s. However, TIR techniques perform much better both at daytime and nighttime [98]. Some modern satellites provide ready-to-use TIR-based products which can be used to detect SDS very easily. For example, METEOSAT SEVIRI-MSG provides RGB Composites for Dust, which indicates any SDS with higher red fraction in the day/night images [171].

3.5.1.3 Microwave and radar imaging

Microwave and radar imaging is also very helpful in detecting SDS, especially in cloudy weather. Since microwave and radar can easily penetrate ice, clouds, etc., they are useful in detecting dust covered by clouds. The microwave-based SDS detection uses the polarization difference technique, which is also known as microwave polarized BT

or microwave polarized index (MPI). For example, data from AMSR-E (Advanced Microwave Scanning Radiometer) onboard Aqua satellite have been used to distinguish dust particles from background desert signature over the Taklamakan desert [190], and Asian dust have been detected with the higher accuracy of about 90% using MODIS by applying infrared BTD and AMSR-E data by applying polarized BTD or MPI [97].

Similarly, TOMS (Total Ozone Mapping Spectroradiometer) Aerosol Index and AMSU (Advanced Microwave Sounding Unit) brightness temperature have been used to detect dust storm over Indo-Gangetic basin [99], and SVM have been used to correctly distinguish dust aerosols from clouds using data from three different satellite devices including Cloud Aerosol Lidar and Infrared Pathfinder Satellite Observations (CALIPSO), MODIS imagery from Aqua satellite, and Cloud Profiling Radar (CPR) from CloudSat satellite [95].

3.5.1.4 Multi-spectral approach

The images in visible spectrum require only simple image processing techniques, such as pixel or edge detection, in order to detect sand and dust events. However, this task becomes challenging in the presence of clouds, sea and surface in the images, and at the nighttime when there is no clear picture available. Fortunately, many satellites produce multi-spectral images, including optical/visible/red-green-blue (RGB), near-infrared (NIR), middle infrared (MIR), and far infrared (FIR) or thermal, and microwave or radar [96]. A multi-spectral image consists of many monochrome images of the same view that are captured by different sensors, and provides extra information that cannot be captured by optical sensors or human eyes. However, processing of the multi-spectral

images requires more computation time and memory [191]. For each spectral band, the wavelengths and applications are shown in Table 3.4.

TABLE 3.4: Spectral bands used by Satellite Imagers

Spectral band	Wavelength (app.)	Typical Applications
Blue	0.4 μm – 0.5 μm	Imaging of earth-atmosphere and deep-water (up to 46 m)
Green	0.5 μm - 0.6 μm	Imaging of vegetation and deep-water (up to 27 m)
Red	0.6 μm – 0.7 μm	Imaging of soil, objects, vegetation and deep-water (up to 9.1 m)
Near infrared (NIR)	0.7 μm – 1.5 μm	Imaging of vegetation
Mid-infrared (MIR)	1.5 μm – 5.6 μm	Imaging of vegetation, soil moisture and fires.
Far infrared (FIR) or Thermal infrared (TIR)	5.6 μm – 1,000 μm	Imaging of fires, dust, geo-structures, and night-time imaging.
Microwave	0.1 cm to 1.0 m	Mapping of terrain and detection of objects.
Radar or Radio	1.0 m to 1 km	Detection of objects.

Multi-spectral approach, also known as integrated approach or multi-sensor approach, combines observations from many satellite sensors including visible, infrared and microwave devices so as to significantly improve the accuracy of SDS detection by avoiding the weaknesses in them when used alone. Some researches use BTM technique alone while others combine it with different techniques for better performance.

A combination of optical and microwave imaging has been successfully employed to monitor dust storms over Nile Delta using MODIS and TRMM Microwave Imager (TMI) data [173]. For the detection of dust storm over Indo-Gangetic basin, TOMS Aerosol Index and AMSU brightness temperature have been successfully used [99]. Similarly, multi-spectral MODIS images have been employed to detect dust storms in North China, both at daytime and nighttime, using TIR method [98]. The BTM technique has been used to detect Asian dust, with an accuracy of about 90%, using data from MODIS and AMSR-E, where visible and infrared observations were employed for SDS detection in clear weather using infrared BTM, while microwave observations were employed to detect SDS below clouds using microwave polarized BTM or MPI [97].

Recently, TOMS and OMI AI (Ozone monitoring instrument aerosol index) satellite data have been used to study dust storms and their horizontal dust loading in the Sistan region [192]. Similarly, MODIS and TOMS have been used to detect transport pathways of Sahara dust over Athens [193].

For multi-spectral images, DIP and AI techniques can be used as well. For example, the maximum likelihood classifier (MLC) and probabilistic neural network

(PNN) techniques are proposed to automate the SDS detection process, which can improve the performance of classification and allow the real-time processing of MODIS images of 1000 m spatial resolution [194].

3.5.2 Unmanned Aerial Vehicle (UAV)

The use of UAVs in SDS detection and other hazards has increased significantly in recent years. UAVs are usually equipped with multiple sensors, including digital cameras for high-resolution imaging, multi-spectral sensors, radar, etc. [195]. Recently, several methods have been developed for the processing of digital images captured by UAVs [196–198], multi-spectral images [195], video flow [199], etc.

UAVs can be deployed quickly over a disaster area and can provide very high-resolution images at relatively low-cost compared to manned aircrafts. However, there is very high cost associated with the initial acquisition of a UAV, crew training and permission from aviation authority [200].

Unfortunately, the application of UAVs in SDS monitoring is very limited due to problems in flight during storms. However, UAVs have been used for the study of aerosol-dust-cloud [201], and a global profiling system has been developed for improved weather and climate prediction using high-altitude long-endurance UAVs [202].

3.6 Hybrid Approaches

Hybrid approaches unite the power of two different approaches and are very useful in detecting all types of SDS, i.e. short-term, small-scale, long-term, and large-scale. They are also suitable for the study of dust transport over long distances.

Hybrid approaches make use of both ground-based and spaceborne observations for efficient detection of SDS. Generally, they combine satellite imaging with another technology such as video surveillance, sensory information, UAVs, etc. For example, a combination of spaceborne observations from satellite and ground-based observations from digital camera stations is very common [127]. The ground-based observations are used to detect short-term or small-scale SDS and capture the startup phase of large-scale SDS, while satellite imaging is used to detect large-scale SDS. The main advantage of such a hybrid approach is that video surveillance works even during the cloudy days when satellite imaging cannot do well. The camera stations are usually used to augment remote sensing satellites in order to enhance their capability of SDS detection.

Similarly, a combination of satellite imaging along with regional dust models, particle samplers and lidar systems on the ground has been used to monitor a severe dust storm over Greece [101]. Moreover, a combination of GIS and WSNs has been used to create pollution maps for urban environment, where the sensor nodes sense and measure pollutants, while GIS provides geographic information. The proposed system creates a map of the pollution for an area and highlights the areas with higher pollution [203].

Hybrid approaches consisting of satellite imaging and WSNs are becoming popular due to their efficiency and effectiveness in detecting SDS of all types [102].

3.7 Comparison of Different Technologies

Table 3.5 provides a quick comparison of different technologies for SDS detection. These technologies are compared in terms of their approach, spatial coverage, temporal coverage, early warning delay, initial cost, operational cost, accuracy, effectiveness, availability, functionality, data from dust origin, and suitability for SDS detection. Accordingly, hybrid approaches are best suitable for detecting SDS of all types.

Satellite imaging has become the most popular technology for SDS detection mainly due to low cost and high availability, while lookout-towers have become obsolete nowadays. Although the cost associated with designing, launching and operating a satellite is very high, most of the satellites make their data freely available to everyone. Satellite imaging can also be used for dust modeling and forecasting as long-term satellite data on dust storm climate is available [123]. However, satellite imaging has some very serious limitations, which include the following:

- 1) ***Poor spatial resolution:*** The spatial resolution of most satellites is very low, which makes it difficult to detect small-scale SDS using satellite imagery. Although some satellites, such as IKONOS, Earlybird and OrbView-3, provide very high spatial resolution, their temporal resolution is too low to detect short-term SDSs.

- 2) ***Poor temporal resolution:*** The temporal resolution of most satellites is not good enough to provide a real-time monitoring of SDSs. For example, GOES satellite provides one of the best temporal resolution (i.e. 15 min), which is still not helpful in detecting short-term SDS.
- 3) ***Inaccurate data on dust and land-surface properties:*** Satellite imaging cannot provide accurate data regarding land-surface and atmosphere for dust modeling etc., which results in low accuracy of SDS prediction [126]. As satellites sense signals that are blend of radiation from several sources including land-surface, dust, clouds and other aerosol, it is not yet possible to use satellite signals for accurate measurement of dust-related physical quantities, such as dust particle size, dust emission rate, etc.

These limitations of satellite imaging are effectively overcome by other technologies such as video-surveillance, sensory information, UAV, etc. These technologies can accurately measure the land-surface properties by deploying a network of sensors/devices in a certain area, and provide more accurate, up-to-date and detailed data on dust, which can be helpful in modeling dust and detecting SDS at very early stage in its origin. The same network of sensors/devices can be used for many other purposes such as detecting fire, rain, snow, volcanic eruption, etc. Although, these technologies have high initial cost, they capture first-hand and accurate data which can be used to detect any kind of dust event except large-scale SDS. However, these technologies have their own limitations, which include the following:

- 1) ***Narrow spatial coverage:*** Due to narrow spatial coverage, it is difficult to detect large-scale SDS using these technologies.
- 2) ***Unavailability of long-term data:*** The prediction of SDS becomes a challenging task due to unavailability of long-term data and broad coverage in these technologies.
- 3) ***High cost and efforts:*** The cost and efforts to establish and maintain systems based on these technologies is very high compared to satellite imaging which is freely available to everyone.

Nevertheless, these limitations of video-surveillance, sensory information and UAVs can be overcome by using hybrid approaches which combine satellite imaging with some of these technologies. This fusion combines the benefits of all technologies employed and makes hybrid approach powerful enough to detect all types of SDS more accurately and efficiently.

A quick comparison of different technologies for SDS detection is provided in Table 3.5, where ‘suitability for SDS detection’ is determined by several factors, such as coverage, efficiency, accuracy, cost, effectiveness, availability, versatile functionality, ability to collect the required data from certain region, etc. Accordingly, hybrid approach is the best suitable technique for SDS detection.

TABLE 3.5: Comparison of Different Technologies for SDS Detection

Technology	Spatial Coverage	Temporal Coverage	Early warning delay	Initial Cost	Operational Cost	Accuracy	Effectiveness	Availability	Functionality	Data from dust origin	Suitability for SDS detection
Lookout- Tower	Small- scale	Short- term and Long- term	Few seconds	High	Low	Low	Low	Low	Limited	Yes	Poor
Video- Surveillance	Small- scale	Short- term and Long- term	Few seconds	High	Low	Medium	Medium	High	Limited	Yes	Fair

Sensory Information	Small- scale to medium- scale	Short- term and Long- term	From less than 1 second to few seconds	High	Low	High	High	High	Versatile	Yes	Good
Satellite Imagery	Large- scale	Long- term only	From more than 15 minutes to many hours or days	*Low	Low	Medium	Medium	High	Versatile	No	Good
Unmanned Aerial Vehicle	Small- scale	Short- term	From few	High	High	Medium	Medium	Low	Limited	Yes	Poor

		and	minutes								
		Long-	to many								
		term	hours								
Hybrid	Small-	Short-	From	Medium	Low	High	High	High	Versatile	Yes	Excellent
Approach	scale to	term	less								
	Large-	and	than 1								
	scale	Long-	second								
		term	to many								
			seconds								

* When satellite images are available on the Internet.

Qualitative rating: poor, fair, good, excellent.

3.8 Conclusions

A brief review of the techniques and technologies for SDS monitoring is provided. As SDS offer severe hazards to the environment, economy and health, it is important to predict any forthcoming SDS in order to alert the general public so that they can take protective actions. A system for detection and prediction of SDS needs data on dust and other environmental changes through ground-based and/or spaceborne observations. Satellite imaging has been commonly used for SDS detection since 1970s. However, it has some serious limitations, including poor spatial and temporal resolution, and inability to provide accurate data regarding land-surface and atmosphere. Conversely, these limitations are overcome by other technologies such as video-surveillance, sensory information and UAV. However, they have their own limitations, including narrow spatial coverage, unavailability of long-term data, and high cost and efforts required to establish these technologies.

Hybrid approaches combine satellite imaging with some of the other technologies such as video surveillance, UAV, sensory information, etc. The use of WSNs in hybrid systems is becoming very common nowadays, mainly due to lower cost and higher performance. By using different technologies at the same time, hybrid approaches achieve benefits of all technologies employed. Using hybrid approaches, systems can detect all the four types of SDS including short-term, small-scale, long-term and large-scale. Therefore, hybrid approaches are best suited for SDS detection and prediction.

CHAPTER 4.

CLOCK SYNCHRONIZATION

Wireless Sensor Networks (WSNs) need accurate time synchronization for data consistency and coordination. Although the existing algorithms for time synchronization offer very good accuracy, their energy-consumption is high and distant nodes are poorly synchronized. A Recursive Time Synchronization Protocol (RTSP) is proposed which accurately synchronizes all the nodes in a network to a global clock using multi-hop architecture in an energy-efficient way. It achieves better performance due to the MAC-layer time-stamping based on Start of Frame Delimiter (SFD) byte, infrequent broadcasts by a dynamically elected reference node, compensation of the propagation delay and adjustment of the timestamps at each hop, estimation of the relative skew and offset using least square linear regression on two data points (2LR), adaptive re-synchronization interval, aggregation of the synchronization requests, and energy-awareness. A detailed analysis of the sources of errors is also provided. Simulation results show that the RTSP can achieve an average accuracy of 0.3 microseconds in a large multi-hop flat network while using 5-times lesser energy than that of FTSP in the long run; and performs even better in a clustered network where it can achieve an average accuracy of 0.23 microseconds while using 7-times lesser energy.

4.1 Introduction

Wireless Sensor Networks (WSNs) require accurate time synchronization, usually less than one microsecond, for many reasons, such as precise time-stamping of messages, in-network signal processing, time-based localization, TDMA-based medium access, cooperative communication, coordinated actuation, and energy-efficient duty cycling of sensor nodes. There are several ways to achieve synchronization, which include ordering of events, local synchronization, global synchronization, etc. However, many applications need global time synchronization which requires all nodes to be synchronized with one global clock.

The issue of clock synchronization has been investigated extensively and several methods or protocols have been proposed for global time synchronization, such as GPS-based clock synchronization, Network Time Protocol (NTP) [204], Precision Time Protocol (PTP) [205], Reference Broadcast Synchronization (RBS) [206], Timing-sync Protocol for Sensor Networks (TPSN) [207], and Flooding Time Synchronization Protocol (FTSP) [208].

The GPS-based approach can provide an accuracy of $1\mu\text{s}$ or better, but it is costly, energy-inefficient and infeasible in obstructed environments. The NTP [204] is commonly used in traditional computer networks including the Internet, but it is not suitable for WSNs because of its very low accuracy (tens of milliseconds), high complexity and energy inefficiency. The PTP, defined by IEEE 1588 standard, can achieve clock accuracy in the sub-microsecond range for networked measurement and

control systems [209–211], but it is suitable only for hierarchical master-slave architecture. The RBS uses receiver-receiver synchronization to produce an average accuracy of $29.1\mu\text{s}$ for a single hop network, but this accuracy is not sufficient for WSNs. The TPSN uses sender-receiver synchronization and MAC-layer time-stamping of messages at the sender side to provide an average accuracy of $16.9\mu\text{s}$ for a single hop network and less than $20\mu\text{s}$ for multi-hop network, but it is energy-inefficient and insufficiently accurate.

The Flooding Time Synchronization Protocol (FTSP) [208] is the most commonly used protocol for clock synchronization in WSNs. It broadcasts messages with timing information from a single sender to several receivers without any exchange among themselves. It dynamically elects a reference node which regularly floods its current timestamp into the network creating an ad-hoc tree structure of the network instead of a fixed spanning tree. The MAC-layer time-stamping at both sender and receiver sides eliminates all kind of random delays except propagation delay. The timestamps are made at the end of each byte after SFD (Start of Frame Delimiter) or SYNC byte, normalized, averaged, error corrected, and then final timestamp is embedded into the message. A node waits for sufficient data points that are pairs of global and local timestamps and then estimates the offset and skew using least square linear regression (LSLR). Any node that is fully synchronized with the reference node begins flooding its own estimation of the global clock. In this way, the FTSP provides an accuracy of $1.48\mu\text{s}$ for single hop case and about $0.5\mu\text{s}$ per hop in a multi-hop network.

There have been many efforts to improve the FTSP in terms of accuracy, efficiency, energy consumption, etc. For example, D. Cox et al. [212] have improved the accuracy and power consumption in a single hop network using a different method of time-stamping that is based on the Start of Frame Delimiter (SFD) byte. In the SFD-based time-stamping, messages are time-stamped using the time at which radio chip generates an interrupt for the microcontroller after the SFD byte has been sent or received. Similarly, M. Aoun et al. [213] have obtained an accuracy of $0.4\mu\text{s}$ in a single hop network by using SFD-based time-stamping along with Kalman-filter for skew estimation. Although the accuracy of FTSP and its improved versions is sufficiently good, the energy consumption is very high and the distant nodes are poorly synchronized.

A Recursive Time Synchronization Protocol (RTSP) [214] have been proposed and presented at the IEEE SAS2012 in Brescia, Italy. The RTSP provides an average accuracy of $0.3\mu\text{s}$ per-hop in a large multi-hop flat network (i.e., without clustering) while using only 1/5th of the energy consumed by FTSP in the long run. However, as WSNs are usually clustered and hierarchical, the RTSP algorithm have been modified and extended to make it work with clustered networks as well. This chapter extends the RTSP algorithm for clustered networks. In case of non-clustered or flat network, each node is assumed to be a clusterhead in order to run RTSP algorithm correctly. Simulation results show that the extended RTSP algorithm further improves the accuracy and energy consumption, i.e., it can provide an average accuracy of $0.23\mu\text{s}$ in a large multi-hop clustered network while using only 1/7th of the energy consumed by FTSP in the long run.

The rest of this chapter is organized as follows. Section 4.2 describes the system model while section 4.3 explains the RTSP algorithm in detail. Section 4.4 analyses the sources of errors, while section 4.5 analysis the sources of efficiency. The performance evaluation of RTSP compared to existing algorithms is discussed in section 4.6. Finally, section 4.7 concludes the chapter and describes the possible future work.

4.2 System Model for RTSP

First, the system model, assumptions, time-stamping, message structure, and recursion in RTSP are briefly described.

4.2.1 Clock Model

Each sensor node has a hardware clock consisting of timer circuitry, which is usually based on quartz crystal oscillator and hence unstable and error-prone. The hardware clock can be translated into logical clock which is used for time keeping.

The hardware clock with instantaneous oscillator frequency $f_i(t)$ at an ideal time t , is defined as (1), where f_0 is the ideal frequency, Δf is the frequency offset, d_f is the drift in frequency, and $r_f(t)$ is the random error process:

$$f_i(t) = f_0 + \Delta f + d_f t + r_f(t) \quad (1)$$

Assuming $t=0$ as the initial reference time, the related logical clock reads time $C_i(t)$ at ideal time t , defined as:

$$C_i(t) = C_i(0) + \frac{1}{f_0} \int_0^t f_i(t) dt \quad (2)$$

The expression can be obtained for the time $C_i(t)$ of clock i at a given ideal time t by combining (1) and (2), as follows:

$$C_i(t) = C_i(0) + \frac{1}{f_0} \int_0^t (f_0 + \Delta f + d_f t + r_f(t)) dt \quad (3)$$

After simplification of equation (3) and elimination of the last two terms representing random clock error and frequency drift respectively,

$$C_i(t) = C_i(0) + (1 + \frac{\Delta f}{f_0})t$$

is obtained, which can be written in very simple form as (4),

where α_i is the clock offset (i.e., the deviation from ideal time) at the reference time $t=0$ and β_i is the clock skew (i.e., the frequency of a clock):

$$C_i(t) = \alpha_i + \beta_i t \quad (4)$$

The clocks of any two perfectly synchronized nodes may drift away from each other at a rate at most 2ρ , where ρ is the rate of change of frequency with respect to the ideal clock. Therefore, in order to keep their relative offset within δ seconds all the time, they need to be resynchronized within $\delta/2\rho$ seconds, i.e., re-synchronization interval $T \leq \delta/2\rho$. However, frequent re-synchronizations are not feasible due to inefficiency, cost, etc. Therefore, it is essential to estimate the drift or skew in order to keep the sensor clocks synchronized without any need for frequent re-synchronizations.

4.2.2 Assumptions

Following assumptions, which are realistic and commonly found in literature, are made in the proposed algorithm:

- 1) Sensor nodes are uniquely identified by their numeric IDs from 0 to $n-1$.
- 2) Time-stamping of messages at MAC-layer is possible for each node.
- 3) Neighboring nodes can communicate over an unreliable and error-corrected wireless channel.
- 4) Broadcasting of messages is possible in the network.
- 5) Skew and connectivity don't change during the short interval between synchronization request and reply.
- 6) Propagation delay in one direction is exactly equal to the other.
- 7) A simple linear relationship exists between the clocks of two sensor nodes in a short duration.

4.2.3 Time-stamping

MAC-layer time-stamping of messages [207], [212], [213], [215], both at sender and receiver sides, is very useful for time synchronization in WSNs. Figure 4.1 shows that the RTSP algorithm, like D. Cox et al. [212] and M. Aoun et al. [213], timestamps messages when radio chip generates an interrupt for the microcontroller after the SYNC/SFD byte has been sent or received. Ignoring the propagation delay, this interrupt occurs almost simultaneously both at the sender and receiver chips with a difference of

3 μ s. Therefore, the timestamp at sender side is incremented by 3 μ s.

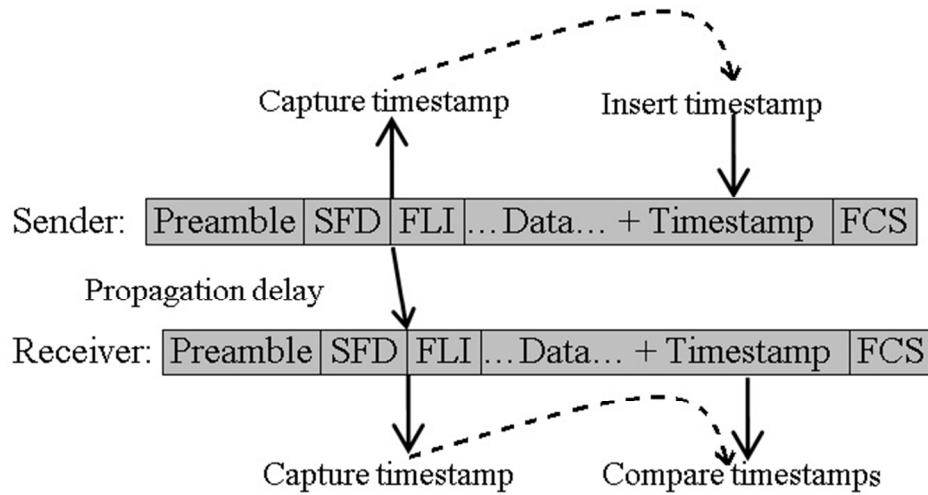


Figure 4.1: Structure of IEEE 802.15.4 MAC frame and the time-stamping

This approach is simple, faster and more accurate than that of FTSP. It is important to note that this approach is also unable to compensate the propagation delay. Therefore, a mechanism for finding and compensating the propagation delays is introduced in order to achieve higher accuracy of time synchronization in the RTSP algorithm.

4.2.4 Structure of the RTSP messages

The RTSP algorithm achieves time synchronization by exchanging messages among nodes. The structure of an RTSP message is shown in Table 4.1. Note that an RTSP-ERN message with a negative value for the refNodeID field is treated as an enquiry, while a non-negative value is treated as an announcement or contest. A node can request timing information by sending an RTSP-REQ message to the reference node,

while the reference node or any synchronized node on the request-path can reply with an RTSP-REP message. The fields T1, T2, T3 and Tr are used for timestamps after SFD byte has been sent, received or forwarded.

TABLE 4.1: Structure of an RTSP Message

Field	Description
msgType	Type of the message: ERN (enquiry/election of the reference node), REQ (request for time synchronization), and REP (reply for time synchronization).
msgID	ID of the message.
originID	ID of the node that originated the message.
imSrcID	ID of immediate source node that forwarded the message.
imDestID	ID of immediate destination for message forwarding.
refNodeID	ID of the reference node known (-1 for ERN enquiry).
T1	Local time when the message was sent or forwarded.
T2	Local time when the message was received.
T3	Local time when reply was sent or forwarded.
Tr	Reference time when the reply was sent or forwarded.

Note that RTSP uses 0 for Null value, and * for broadcast address, and that RTSP-REQ message takes Null values for T2, T3 and Tr.

4.2.5 Recursion and Multi-hop Synchronization

The RTSP algorithm does not require any tree-like structure; it works with any network topology, flat and clustered. A time synchronization request can be initiated by any node in the network, which is recursively forwarded to the reference node. The reference node, or any synchronized node on the request-path, replies with timing information that is sent back to the requesting node through the same path. It is important to note that the propagation delay is also compensated at each hop before forwarding the reply.

Figure 4.2 explains the request-and-reply mechanism in RTSP. The node A in a multi-hop network sends request at T_1 to the reference node via another node B that receives it at time T_2 . The node B, if not synchronized, forwards this request to another node in a recursive manner until it reaches the reference node or a fully synchronized node N which replies with the timing information that is forwarded to node B through the same path. The node B at time T_3 forwards this reply to node A which receives it at time T_4 . S. Ganeriwal et al. [207] have already proved that the performance of TPSN is insensitive to any variations in processing delay, i.e., the interval between T_2 and T_3 does not affect the accuracy of time synchronization. This gives us an opportunity to make recursive calls during the interval between T_2 and T_3 in the RTSP algorithm without losing performance.

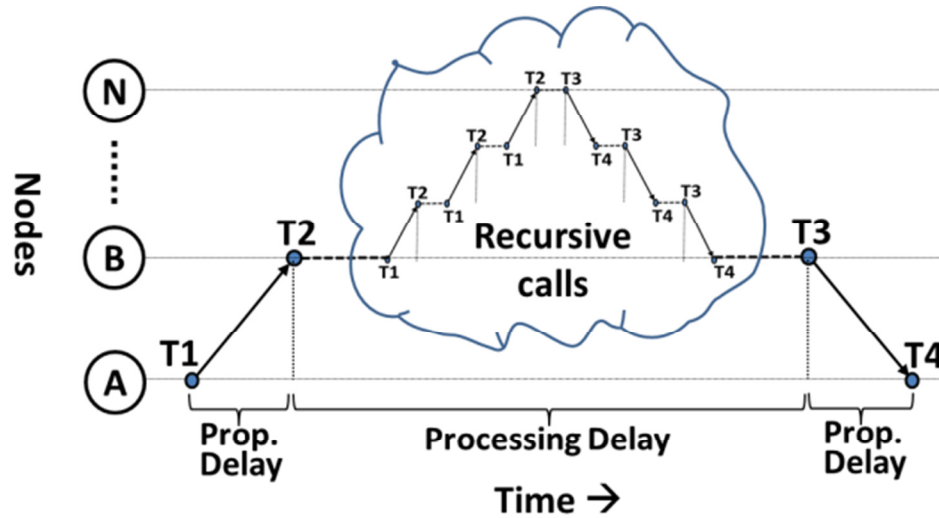


Figure 4.2: The request-and-reply mechanism in RTSP

Moreover, the nodes closer to the reference node are expected to do the following actions before the distant nodes: 1) receive announcement about the newly elected reference node, 2) initiate the time synchronization request on knowing ID of the reference node, and 3) become synchronized with the reference node. As the closer nodes are expected to become synchronized before distant nodes and any synchronized node on the request-path can reply, very few time synchronization requests are forwarded all the way to the reference node.

4.3 The Recursive Time Synchronization Protocol (RTSP)

After a WSN boots up, the sensor nodes broadcast an RTSP-ERN enquiry message to ask their neighbors about the identification of reference node, wait for the period T or until a reply is received, run repeatedly the RTSP algorithm that takes care of the dynamic election of a single reference node with the smallest ID and compensation of

the offset and drift. This process is outlined in the Algorithm 1: RTSP. Each node maintains a few variables including myRef for ID of the reference node, and myClient for IDs of the nodes sending time synchronization request through it.

Note that in case of non-clustered or flat network, each node is assumed to be a clusterhead in order to run RTSP algorithm.

Algorithm 1: RTSP

1. Check if new RTSP message is received (or overheard);
2. // on receiving a new message, do the following
3. **If** a new authenticated message is received **then**
4. **If** msg→msgType == “ERN” **then**
5. T2 = timestamp after SFD byte received;
6. **If** msg→refNodeID<0 AND myRef!=Null **then**
7. // an enquiry; reply if reference ID known
8. T3 = timestamp after SFD byte sent;
9. RTSP(“ERN”, msg→msgID, myID, myID,
10. msg→imSrcID, myRef, msg→T1,
11. T2, T3, Tr);

```

12.      Else if (msg→refNodeID >=0 AND
13.          myRef is empty) OR
14.          myRef > msg→refNodeID then
15.          //reference ID unknown/expired; accept new
16.          myRef = msg→refNodeID;
17.          RTSP("ERN",msg→msgID, msg→originID,
18.              myID,*, msg→refNodeID,0,0,0, Tr);
19.      Else if myID < msg→refNodeID AND I'm a
20.          clusterhead AND myEnergy>low then
21.          //smaller ID & fair energy; enter into contest
22.          myRef = myID;
23.          RTSP("ERN", msgID, myID, myID, *,
24.              myID, 0, 0, 0, Tr);
25.      Else if myRef == msg→refNodeID AND
26.          I'm a gateway or a clusterhead AND
27.          myID != msg→originID AND
28.          msg→msgID is new then
29.          // new msg from reference node; rebroadcast

```

```

30.         RTSP("ERN",msg→msgID, msg→originID,
31.             myID,*,msg→refNodeID, 0,0,0,Tr);
32.     End if
33. Else if msg→msgType == "REQ" then
34.         T2 = timestamp after SFD byte received;
35.         If myID==refNodeID OR I'm a synchronized
36.             gateway or clusterhead then
37.             // reply as reference or a synchronized node
38.             T3 = timestamp after SFD byte sent;
39.             Tr = T3;
40.             RTSP("REP",msg→msgID, msg→originID,
41.                 myID, msg→imSrcID, myID,
42.                 msg→T1, T2, T3, Tr);
43.         Else if I'm a gateway or clusterhead then
44.             //save T1, T2, source, and do a recursive call
45.             T1old = msg→T1;
46.             T2old = T2;
47.             myClient = msg→imSrcID;

```

```

48.          T1 = timestamp after SFD byte sent;
49.          RTSP("REQ",msg→msgID, msg→originID,
50.              myID, myRef, myRef, T1, 0, 0, Tr);
51.      End if
52.  Else if msg→msgType == "REP" then
53.      // calculate d, and adjust and record the timestamp
54.      T4 = timestamp after SFD byte received;
55.      d = ((msg→T2–msg→T1)+(T4–msg→T3))/2 ;
56.      Tr = msg→Tr + d;
57.      Record timestamps global=Tr & local=T4;
58.  If myID != msg→originID AND
59.      myClient not empty then
60.      // recover T1 & T2; and forward the msg
61.      T1 = T1old; T2 = T2old;
62.      T3 = timestamp after SFD byte sent
63.      Tr = Tr+(T3–T4); //+ elapsed time since T4
64.      RTSP("REP",msg→msgID, msg→originID,
65.          myID, myClient, msg→refNodeID,

```

```

66.          msg→T1, T2, T3, Tr);

67.          Clear the variable myClient;

68.      End if

69.  End if

70. End if

71. //ask the id of reference node if not known already

72. If myRef == Null AND I'm a clusterhead then

73.    RTSP("ERN",msgID,myID,myID,*,-1,T1,0,0, Tr);

74. Else if myRef == Null AND I'm not clusterhead then

75.    RTSP("ERN", msgID, myID, myID, myCH,

76.        -1, T1, 0, 0, Tr);

77. End if

78. // if no msg received for T, start contest for reference

79. If no ERN msg received for T AND I'm clusterhead

80.    AND myID!=myRef AND myEnergy>low then

81.    // contest for reference node by broadcasting ERN

82.    myRef = myID;

83.    RTSP("ERN", msgID, myID, myID, *, myID, 0,

```



```

84.          0, 0, Tr);

85. End if

86. //a reference node broadcasts every after T period

87. If myID == myRef AND didn't broadcast for T AND

88.      myEnergy > low then

89.      // an active reference node rebroadcasts after T

90.      RTSP("ERN", msgID, myID, myID, *, myID, 0,

91.          0, 0, Tr);

92. End if

93. // if out of synchronization, send synchronization request

94. If ((new REP msg overheard OR ERN msg received)

95.      AND myRef == msg→refNodeID AND

96.      local time much different than msg→Tr) OR

97.      skew is much lesser or greater than 1 then

98.      // node is out of synchronization; so send request

99.      RTSP("REQ", msgID, myID, myID, myRef,

100.         myRef, T1, 0, 0, Tr);

101. End if

```

```

102. // Estimate offset & drift using two data points

103. If two data points, with one new, are available then

104.     Calculate  $\alpha$  and  $\beta$  using the last two data points;

105.     Update the local clock to become synchronized;

106. End if

```

The RTSP algorithm is responsible for two major functions, 1) election of the reference node, 2) compensation of the offset and drift, which are explained in the next two sub-sections.

4.3.1 Election of the Reference Node

The RTSP algorithm dynamically elects a single reference node as follows:

- 1) After start up, a sensor node waits for short time, and then sends an RTSP-ERN message containing ‘-1’ value for the refNodeID field in order to ask the neighboring nodes for ID of the reference node (Algorithm 1: RTSP, line 71–77). It is important to note that a clusterhead broadcasts the enquiry message, but a non-clusterhead sends the enquiry message to its own clusterhead. The enquirer node waits for the duration T or until a reply is received, and then acts as follows.

- a) If it does not receive any reply, it enters into the contest for new reference node by broadcasting an RTSP-ERN message by putting its own identification in the refNodeID field (Algorithm 1: RTSP, line 78–85). Note that a node that is not a clusterhead or is low in energy cannot take part in the contest for reference node.
 - b) However, if it receives a reply, it saves the identification of reference node in a local variable called myRef, and then broadcasts the RTSP-ERN message (Algorithm 1: RTSP, line 12–18).
- 2) If a new RTSP-ERN message is received and authenticated (Algorithm 1: RTSP, line 1–4), it checks the value of refNodeID field to do the following.
 - a) If the refNodeID field contains any negative value, the message is treated as an enquiry. If a node knows the identification of reference node, it replies directly to the enquirer (Algorithm 1: RTSP, line 6–11).
 - b) However, if the refNodeID field contains any non-negative value, the message is treated as an announcement or contest, and is flooded as follows.
 - i) If the receiving node does not know the identification of reference node or it knows some identification that is greater than refNodeID, it learns the identification of new reference node by updating myRef variable and then rebroadcasts the message (Algorithm 1: RTSP, line 12–18).

- ii) If the ID of receiving node is smaller than the current reference node and the receiving node is a clusterhead with sufficient energy, it contests for reference node by broadcasting an RTSP-ERN message (Algorithm 1: RTSP, line 19–24).
- iii) If a clusterhead or a gateway already knows the same ID of reference node, it just rebroadcasts the message (Algorithm 1: RTSP, line 25–32).

The elected reference node takes responsibility of broadcasting the timing-information periodically (Algorithm 1: RTSP, line 86–92). However, the frequency of broadcasting is much lower, i.e., $1/10^{\text{th}}$ of the FTPS because RTSP does not rely on this broadcast for the compensation of skew and offset. Instead, the RTSP algorithm uses this broadcasting mainly for announcing the identification of reference node. Any node that receives a new RTSP-ERN message from the reference node accepts it and then rebroadcasts to its neighbors. However, a duplicate message is ignored or rejected by the receivers.

It is possible that more than one node declare themselves as reference nodes simultaneously. To avoid any conflict, a reference node retreats to an ordinary node on receiving an RTSP-ERN message from another reference node with a smaller identification.

Although an RTSP-ERN message is mainly used to elect and announce the reference node in a network, it is also used to find if a node is losing its accuracy of time

synchronization and should request timing information using the RTSP algorithm (Algorithm 1: RTSP, line 93–101).

4.3.2 Offset & Drift Compensation

In order to compensate for the offset & drift, and to synchronize all the nodes on request-path to the reference node, the RTSP algorithm (Algorithm 1) works as follows.

- 1) A node checks the type of newly received RTSP message.
- 2) If an RTSP-REQ message is received, it notes the receive time T2. If it is a reference node or a synchronized clusterhead or gateway node, it replies with an RTSP-REP message containing timestamps T1, T2, T3 and Tr along with other information (Algorithm 1: RTSP, line 33–42). However, an unsynchronized intermediate node recursively forwards the request to the reference node after saving ID of its client in myClient, and the values of T1 and T2 in T1old and T2old respectively (Algorithm 1: RTSP, line 43–51).
- 3) However, if an RTSP-REP message is received, it calculates the value of propagation delay by equation (5).

$$d = \frac{(T2 - T1) + (T4 - T3)}{2} \quad (5)$$

Then it finds the new value of Tr by adding d to the received value of Tr as given by equation (6).

$$Tr = Tr + d \quad (6)$$

Then it records the global and local timestamps where global=Tr and local=T4. Moreover, if the node is an intermediate node, it retrieves the values of T1 and T2, calculates the value of Tr by adding to it the elapsed time since T4 using (7), and then forwards the reply to its client node, if any (Algorithm 1: RTSP, line 52–70).

$$Tr = Tr + (T3 - T4) \quad (7)$$

Figure 4.3 explains the request-and-reply mechanism of RTSP algorithm with the help of an example clustered network.

In Figure 4.3, node 99 initiates an RTSP-REQ at its local time T1 which is received by an intermediate node 5 at its local time T2. The node 5 stores the identification of node 99 in myClient, values of T1 and T2 in T1old and T2old respectively, and then forwards this request at its local time T1 to node 4 which recursively forwards it to the reference node 0. On receiving the request at its local time T2, the reference node 0 processes the request and sends back an RTSP-REP message at its local time T3. When node 4 receives the RTSP-REP reply at time T4, it performs the following six actions: 1) calculates the value of d, 2) adjusts the timestamps by adding d to Tr, 3) records the global and local timestamps, 4) retrieves the values of T1 and T2, 5) calculates the value of new Tr by adding to it the elapsed time since T4, and 6) forwards the reply to node 5 at its local time T3. Node 5 also performs the six actions and then

recursively forward the reply to the node 99. Similarly, node 99 performs the first three actions only since it is not an intermediate node. Note that a reply follows exactly the same path as request, but a new request/reply message may follow a different path from the previous message.

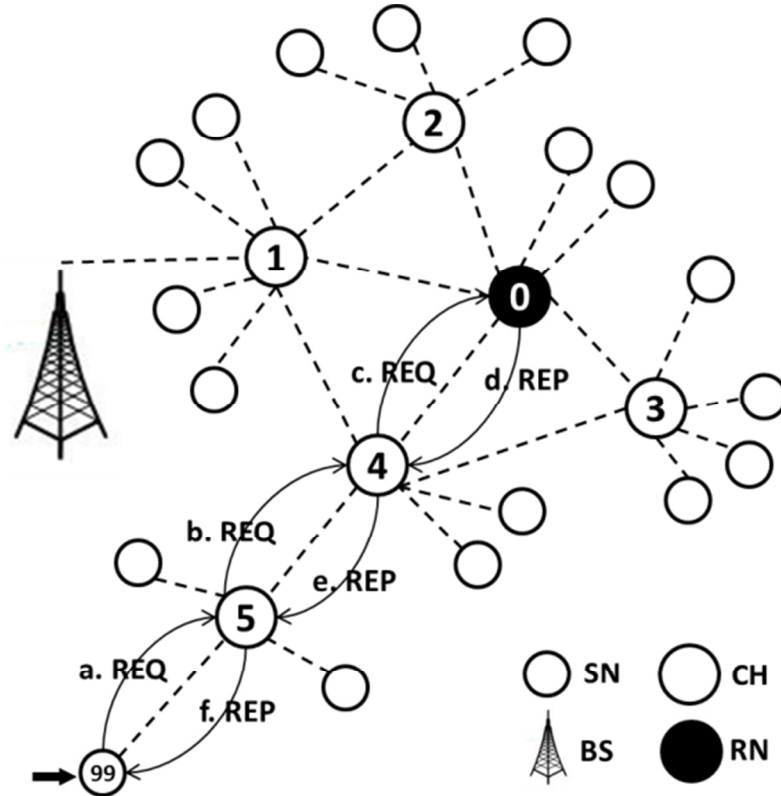


Figure 4.3: Example of request-and-reply mechanism in RTSP algorithm.

The request is recursed only when intermediate nodes are not synchronized. In the next few sections, some unique features of the RTSP algorithm which enhance its performance are described.

4.3.3 Adaptive Re-Synchronization Interval

A simple linear model for clock, as given in equation (4), is not reasonable for long time intervals because of the drastic changes in clock drift due to any sudden change in the environmental temperature, humidity, etc. That is why clock drifts are repeatedly reassessed by re-synchronizing the nodes periodically at a static interval. However, a static period is not feasible for dynamic environments. Therefore, adaptive re-synchronization interval is used in the RTSP algorithm, which allows each individual node to decide if and when to initiate a time synchronization request based on either the value of skew or the deviation of local time from the global time as follows (Algorithm 1: RTSP, line 93–101).

- 1) ***When skew is far away from 1:*** A node is considered to be perfectly synchronized when the current value of skew or drift is 1. Therefore, the value of skew can tell a node if it should initiate a time synchronization request or not. Accordingly, if the current value of skew is much lesser or greater than 1, the node will request timing information by sending an RTSP-REQ message to the reference node; otherwise it will not request any update.
- 2) ***When local time is different from global:*** Although an overheard message is not yet compensated for offset, skew, or propagation delay, however it can provide a rough idea of the global time. On overhearing an RTSP-REP message, a node can check if its local time is almost same as the global timestamp in Tr field. If the two times are

much different, depending on the required accuracy, the node will initiate a time synchronization request; otherwise it will not request any update.

Such an adaptive re-synchronization interval helps reduce the overhead of unnecessary time synchronization requests in the long run. It also provides for the long-term synchronization in a self-configuring way. Besides this, the RTSP algorithm (Algorithm 1) performs a few other functions, which include restarting the process for election of reference node on detecting a failure, broadcasting the RTSP-ERN messages periodically by an active reference node, and calculating the values of α and β for updating the local clock.

4.3.4 Aggregation of the Synchronization Requests

In order to further reduce the number of time synchronization requests and hence energy-consumption, the idea of request aggregation is supported in the RTSP algorithm. Consider the example in Figure 4.3 and assume that another node under clusterhead 5 also initiates an RTSP-REQ message soon after node 99. The intermediate node (i.e., node 5) forwards only the first request to the reference node, waits for a reply, and then forwards this reply to all clients. In order to aggregate synchronization requests, node 5 temporarily saves information on msgID, myClient, T1 and T2 for both requests. It then forwards only one request to node 4. On receiving a reply, it uses the saved information for forwarding the reply to both requesters.

4.3.5 Security

WSNs face many security challenges due to their environment and constraints, which include vulnerability to physical capture, the use of insecure and unreliable wireless channel, and availability of the limited resources such as computing, memory and energy [216]. Moreover, the traditional protocols for time synchronization in WSNs do not consider security and hence prone to several kind of security attacks [217]. Although security is not the prime objective of RTSP, it provides some basic level of protection against the attacks on time synchronization in which a single compromised node may propagate incorrect timing information through the network. Simple techniques such as redundancy, authentication and refusal to forward corrupt synchronization information are used to tackle the attacks on time synchronization as follows.

- 1) **Redundancy:** RTSP algorithm provides some level of redundancy to avoid dependence on just one neighbor. For example, instead of receiving timing information always from the parent node like FTSP, each new request/reply message in RTSP may follow a different path. Also, a node may receive broadcasts by the reference node via a different neighbor at a different time. This provides redundancy for basic level of security.
- 2) **Authentication:** A node that wants to join the network is authenticated first by using any light-weight authentication protocol [216][218]. Moreover, the RTSP algorithm is run only when message is received from an authenticated node (Algorithm 1: RTSP, line 1–3).

- 3) ***Discarding the corrupt information:*** Each RTSP message carries the current or real time of the sender in Tr field, which helps identify any timing information that is very much deviating from the recent data. A node can discard any corrupt message in order to stop the propagation of incorrect timing information through the network.

4.3.6 Energy-awareness and Efficiency

Nodes with different residual energy, i.e., low, medium and high may exist in a WSN at any time. However, low-energy nodes should not contest for the reference node. Moreover, when an active reference node becomes low in energy, it should retreat to an ordinary node and the process for election of the reference node should take place again. The RTSP algorithm ensures that any node contesting for reference node has sufficient residual energy (Algorithm 1: RTSP, line 19–24, and 78–85) so that it can stay active for a long time, broadcast RTSP-ERN messages regularly (Algorithm 1: RTSP, line 86–92), and actively reply to the queries forwarded to it. It will also save on the energy consumed by frequent re-elections of the reference node otherwise. However, when the reference node becomes low in residual energy, it retreats to an ordinary node and the election process takes place again (Algorithm 1: RTSP, line 19–24).

In RTSP, the frequency of broadcasts by the reference node is $1/10^{\text{th}}$ of the frequency in FTSP which saves huge amount of energy. Moreover, adaptive nature of the algorithm reduces unnecessary time synchronization requests and hence saves on energy. Note that once a node becomes synchronized, it can reply to time synchronization requests routed through it without any need to forward them to the reference node. This

reduces burden on the reference node and saves on energy by not forwarding the request all the way to reference node.

4.4 Error Analysis of RTSP

The uncertainties in message delivery include the following time delays: send, access, transmission, propagation, reception, receive, interrupt handling, encoding, decoding and byte alignment time [208][207][215]. Although the RTSP algorithm uses request-and-reply mechanism similar to TPSN but in a recursive manner, it uses MAC-layer time-stamping of messages like FTSP that removes all the sources of uncertainties in message delivery delays except the propagation delay [208]. Therefore, TPSN-like approach is followed in order to analyze the errors of RTSP algorithm.

Consider the request-and-reply mechanism of RTSP (as in Figure 4.2) where node A sends a request at T_1 to another node B that receives it at time T_2 according to their local clocks. However, the real time measured by an ideal clock at these nodes is denoted by t_1 and t_2 respectively. The following equation was derived for TPSN in [207]:

$$t_2 = t_1 + S_A + P_{A \rightarrow B} + R_B \quad (8)$$

where S_A is the time taken to send message (send + access + transmission time) at node A, R_B is the time taken to receive the message (reception + receive time) at node B, and $P_{A \rightarrow B}$ is the propagation time between nodes A and B.

As the MAC-layer time-stamping of messages removes all the sources of uncertainties except the propagation delay [208], equation (8) can be written as follows, for RTSP:

$$t_2 = t_1 + P_{A \rightarrow B} \quad (9)$$

The corresponding equation in terms of local clocks, which involves clock drifts as well, can be written as follows:

$$T_2 = T_1 + P_{A \rightarrow B} + D_{t1}^{A \rightarrow B} \quad (10)$$

Now, the relative drift between the two nodes from $t1$ to $t4$, i.e., $RD_{t1 \rightarrow t4}^{A \rightarrow B}$, is the difference of their respective drifts, which can be positive or negative depending on the value of two drifts, and is given by the following equation:

$$RD_{t1 \rightarrow t4}^{A \rightarrow B} = D_{t1}^{A \rightarrow B} - D_{t4}^{A \rightarrow B} \quad (11)$$

From equation (11), the value of $D_{t1}^{A \rightarrow B}$ can be obtained and put in equation (10) to get the following equation:

$$T_2 = T_1 + P_{A \rightarrow B} + D_{t4}^{A \rightarrow B} + RD_{t1 \rightarrow t4}^{A \rightarrow B} \quad (12)$$

After receiving the request message at time $T2$, node B forwards this request to another node in a recursive manner until it reaches the node N, which replies with timing information that is forwarded to node B through the same path. The node B at time $T3$

forwards this reply to node A which receives it at time T4 according to their local clocks.

Using the analysis similar to (10), following equation can be derived:

$$T_4 = T_3 + P_{B \rightarrow A} + D_{t3}^{B \rightarrow A} \quad (13)$$

Note that $D_{t3}^{B \rightarrow A} \approx D_{t4}^{B \rightarrow A}$ and $D_{t4}^{B \rightarrow A} = -D_{t4}^{A \rightarrow B}$. Hence, equation (13) can be written as follows:

$$T_4 = T_3 + P_{B \rightarrow A} - D_{t4}^{A \rightarrow B} \quad (14)$$

Subtracting (14) from (12) gives the following equation:

$$T_2 - T_4 = T_1 + P_{A \rightarrow B} + D_{t4}^{A \rightarrow B} + RD_{t1 \rightarrow t4}^{A \rightarrow B} - T_3 - P_{B \rightarrow A} + D_{t4}^{A \rightarrow B} \quad (15)$$

Re-arranging equation (15), and substituting $(P_{A \rightarrow B} - P_{B \rightarrow A})$ with P^{UC} gives equation (16), where P^{UC} represents the uncertainty in propagation time:

$$(T_2 - T_1) - (T_4 - T_3) = P^{UC} + RD_{t1 \rightarrow t4}^{A \rightarrow B} + 2D_{t4}^{A \rightarrow B} \quad (16)$$

As offset $\Delta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2}$, therefore equation (16) can be written as follows:

$$2\Delta = P^{UC} + RD_{t1 \rightarrow t4}^{A \rightarrow B} + 2D_{t4}^{A \rightarrow B} \quad (17)$$

To correct the clock at T4 at node A, subtract the value of $D_{t4}^{A \rightarrow B}$ from Δ , which results in the following equation:

$$Error = \Delta - D_{t4}^{A \rightarrow B} = \frac{P^{UC}}{2} + \frac{RD_{t1 \rightarrow t4}^{A \rightarrow B}}{2} \quad (18)$$

From (18), it can be concluded that the RTSP algorithm has only two sources of error:

- 1) ***Variation in Propagation Delays (P^{UC})***: The RTSP algorithm assumes that the propagation delay in one direction is exactly equal to the other. When this is true, one source of synchronization error is eliminated and accuracy of the time synchronization is improved further. However, this assumption may not be true in wireless communication. Consequently, each hop might add possibly an inaccurate value of d to the reference time Tr in order to find the new value of Tr (Algorithm 1: RTSP, line 56). Fortunately, the variation in propagation delays can be neglected. Since propagation delay is calculated as distance between two nodes divided by the speed of radio signal through air that is close to speed of light, the variation in propagation delays can be neglected because the change in two determining factors is negligible during the short interval between T1 and T4. For example, the maximum increase in distance between two nodes moving at the speed of 1000km/h will be only 0.0006m/ μ s. Similarly, the speed of radio signal affected by changes in the air temperature and humidity is negligible for a distance of hundreds of miles; rather

totally negligible for a distance of few hundred meters in WSNs. Therefore, it can be concluded that the variation in propagation delays in WSNs is usually negligible.

2) **Relative Drift Between Local Clocks** ($RD_{t1 \rightarrow t4}^{A \rightarrow B}$): The hardware clock in a low-cost sensor node is usually based on quartz crystal oscillator that is vulnerable to huge drift from the ideal clock. However, equation (18) shows that the RTSP is sensitive only to the “relative drift” between the two clocks instead of the drift from ideal clock. It calculates the relative drift (β) and offset (α) for eq. (4) using 2LR as follows in (19-20), where x_i is the global time while y_i is the local time:

$$\beta = \frac{x1 - x2}{y1 - y2} \quad (19)$$

$$\alpha = \bar{y} - \beta \bar{x} \quad (20)$$

The skew or drift between two clocks can be compensated using equations (4), (19) and (20). Consequently, the effect of this source of error is also minimized by the RTSP algorithm.

4.4.1 Propagation of errors in multi-hop

Any error at one hop may increase when it propagates to the next hops. But in case of time synchronization, the error may not always increase due to different sign and magnitude of the drift rate of sensors.

Let, in a network of k hops, every hop may experience a random error of δ . As the sum of randomly distributed errors is the square root of the sum of the squares of errors on each hop, the total error at k^{th} hop may be calculates as:

$$Error_{k-hop} = \sqrt{(\delta_1)^2 + (\delta_2)^2 + \dots + (\delta_k)^2} \quad (21)$$

which can be written in a simpler form as follows:

$$Error_{k-hop} = \delta\sqrt{k} \quad (22)$$

Therefore, the total error at k^{th} hop may be in the order of $\delta\sqrt{k}$ which means that the accuracy of time synchronization does not decrease much with the increase in number of hops.

4.4.2 Bounds on total error in multi-hop

According to (22), the total error at k^{th} hop may be in the order of $\delta\sqrt{k}$. However, due to different sign and magnitude of error at each hop, the errors may sum up to a much smaller value which can be zero as well. Extreme situations occur when all the errors are either negative or positive. Accordingly, the total error at k^{th} hop is $-\delta\sqrt{k}$ for all negative values of δ while $+\delta\sqrt{k}$ for all positive values of δ . This determines the bounds on total error in multi-hop network as:

$$-\delta\sqrt{k} \leq Error_{k-hop} \leq +\delta\sqrt{k} \quad (23)$$

4.5 Efficiency Analysis of RTSP

The RTSP algorithm is fast and efficient in terms of setup time or delay, computation and memory usage, and energy consumption. In this section, the efficiency of RTSP algorithm is analyzed. Figure 4.4 shows that the computation and memory requirements and setup delay of RTSP are much lesser than that of FTSP.

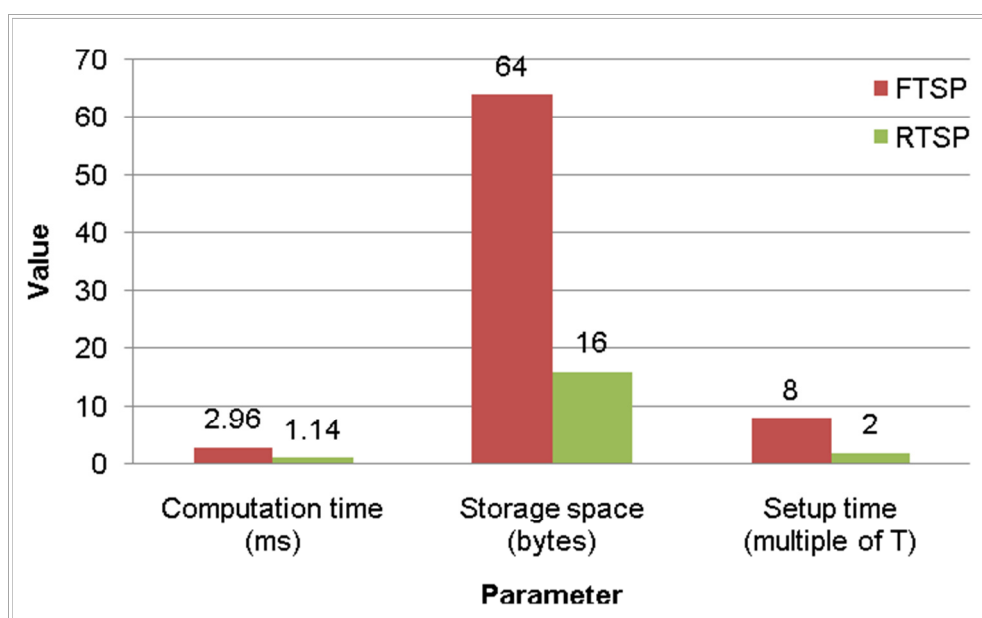


Figure 4.4: Computation and memory requirements of RTSP vs. FTSP

After the network starts up, RTSP can synchronize a node as soon as it receives two data points which can be requested by a node as soon as it knows the ID of reference node. Therefore, the setup time of RTSP is about $2T$ compared to $8T$ for FTSP, where T is the fixed interval for re-synchronization in FTSP.

The time-stamping mechanism of RTSP is much simpler when compared to FTSP. Moreover, RTSP calculates skew and offset using 2LR instead of 8LR in FTSP. Accordingly, RTSP requires much lesser time to execute its steps and lower memory to store the data points.

The RTSP algorithm consumes lesser energy compared to FTSP. It saves energy through infrequent reference broadcasts, and by reducing the number of requests through the adaptive re-synchronization interval and request-aggregation.

4.6 Performance Evaluation

In order to evaluate the performance of proposed algorithm, a simulation in MATLAB is performed using the parameters given in Table 4.2. Note the use of random topology, random walk mobility model and pause time of zero, which makes the network highly dynamic. A time synchronization protocol that performs better in such a network is expected to perform even better in a static environment. The simulation was run to collect data on average synchronization error, per-hop synchronization error, and energy consumption in the long run.

Figure 4.5 shows the average absolute error of time synchronization for different algorithms, which is obtained by running the simulation for thousands of times with different parameters. The average absolute error of RTSP algorithm is $0.288\mu\text{s}$ for flat network and $0.230\mu\text{s}$ for clustered network, which is significantly lower than FTSP

protocol that is $0.485\mu\text{s}$. However, the performance of RBS and TPSN is very bad compared to other algorithms.

TABLE 4.2: Simulation Parameters

Parameter	Description
No. of nodes	50, 100, 200, 500
Prob. of becoming clusterhead	5%
Topology	Random
Deployment area	50m x 50m to 500m x 500m
Mobility	High (pause time 0)
Mobility model	Random Walk
Processing delay (at each node)	1ms – 100ms
No of simulations (results averaged)	1000
Channel	Wireless
MAC	IEEE 802.15.4/ ZigBee
Antenna	Omni
Transmission range	30 m
Routing protocol	AODV

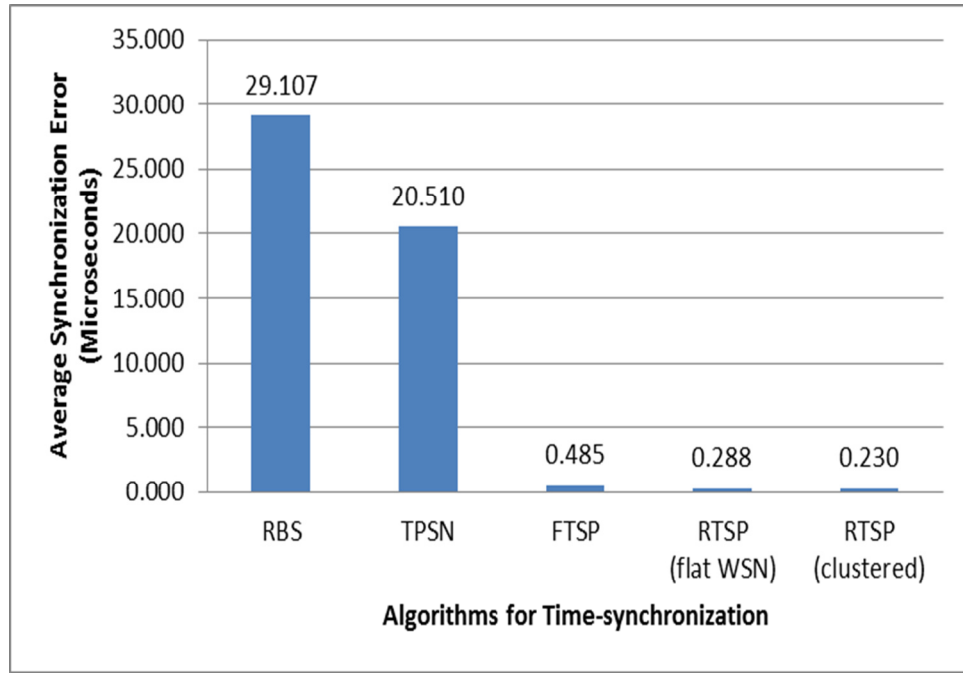


Figure 4.5: Average absolute error of time synchronization

As WSNs are usually multi-hop networks, it has always been interesting to compare the average synchronization error per-hop. Figure 4.6 compares the average synchronization error of RTSP algorithm with TPSN and FTSP. RBS is skipped from the comparison because it is not a multi-hop protocol. The Figure 4.6 shows that average synchronization error of the RTSP algorithm for nodes at 10th hop is 0.420 μ s for flat network and 0.388 μ s for clustered network, which is even lesser than the average synchronization error of FTSP in a 1-hop network. Similarly, Figure 4.7 compares the average synchronization error versus number of nodes in the network; again the RTSP algorithm performs better than others.

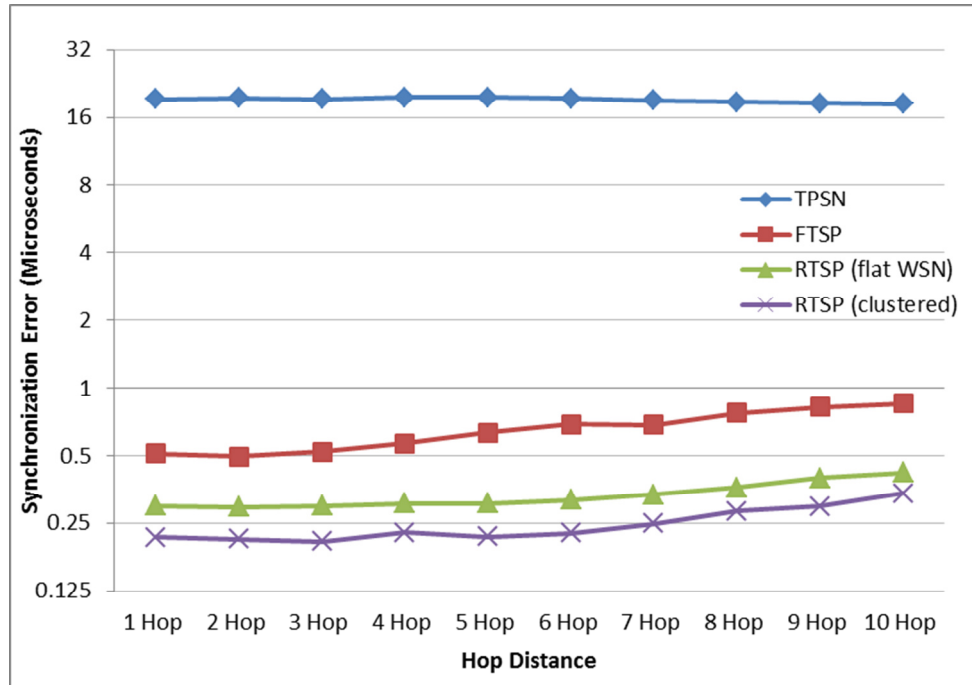


Figure 4.6: Average absolute error of time synchronization (per-hop)

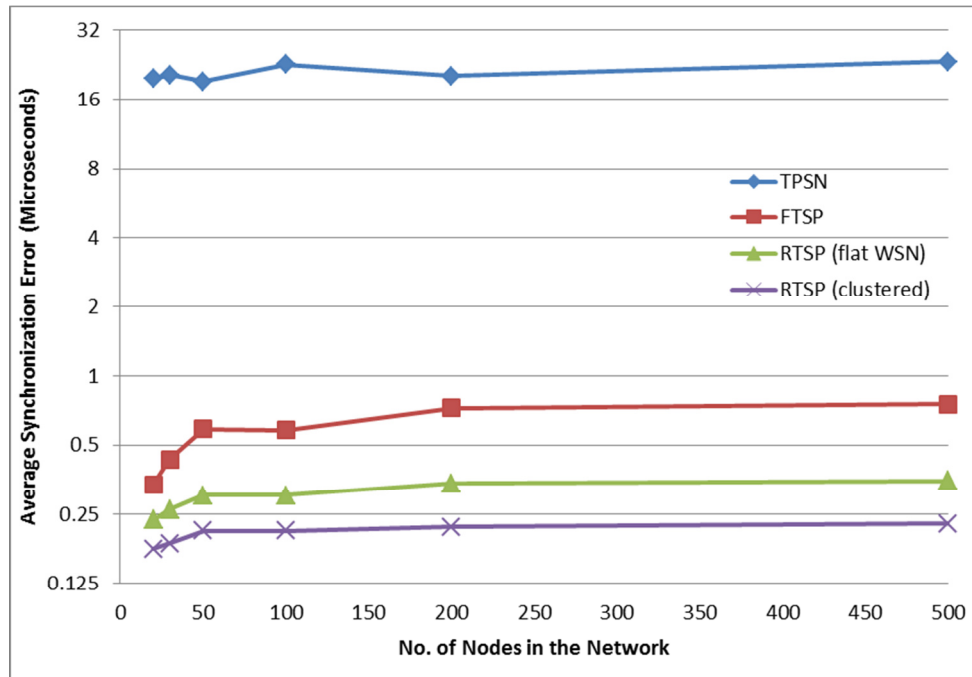


Figure 4.7: No. of nodes vs. average absolute error of time synchronization

Energy consumption is very important factor in the performance of time synchronization algorithms. However, actual energy consumption is affected by many factors, such as type of the hardware, software, and antenna etc. Therefore, the energy consumption is measured in terms of total number of time synchronization messages exchanged among the nodes. If the re-synchronization period is T seconds (based on FTSP) and N is the total number of nodes, then each node sends N messages in RBS protocol, 2 messages in TPSN protocol, 1 message in FTSP protocol [208], and only 0.14 messages in RTSP protocol in the long run. Calculating the total number of messages sent by RTSP is straight forward. On start-up of the network, each node forwards 1 message, which is broadcasted by the reference node. Then a node sends two RTSP-REQ messages and gets their replies accordingly. This makes 5 messages per node. But once the nodes are synchronized, they need to send very less number of requests due to the adaptive behavior of RTSP algorithm. To find the total number of messages sent in RTSP in the long run, data is collected from simulation of a network of 100 nodes for very long time. Figure 4.8 shows that, in the beginning, total energy consumption (i.e. no. of messages exchanged) of the RTSP algorithm is higher than TPSN and FTSP. However, it becomes equal to TPSN after 4 or 5 periods and equal to FTSP after 7 or 8 periods, and then remains always lower than the two protocols in the long run. Using the Curve Fitting Toolbox in MATLAB, the regression equation was found for RTSP curve which revealed that each node sends only 0.2 messages for flat network and 0.14 messages for clustered network in the long run, which means that it consumes 5 to 7 times lesser energy than FTSP.

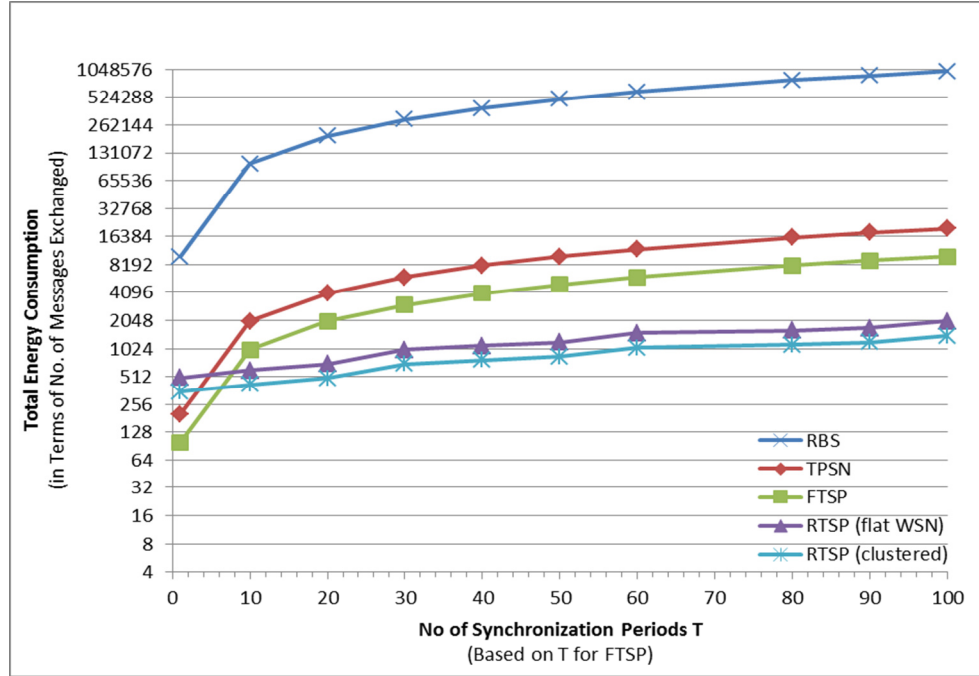


Figure 4.8: Total energy consumption in the long run (for 100 nodes)

Table 4.3 provides a quick comparison of the RTSP algorithm with other global time synchronization algorithms.

4.7 Cost or Limitations of RTSP

The RTSP algorithm has a few limitations which may affect its performance; mainly due to its assumptions. For example, a linear clock model performs well only in short and medium intervals. However, for very long intervals, a sophisticated and complex clock is required. Moreover, due to sudden changes in connectivity, the reply of a time synchronization request may be lost and a new request must be initiated after timeout.

TABLE 4.3: Comparison of RTSP with Other Algorithms

Property	RBS	TPSN	FTSP	RTSP
Scheme used	Receiver-to-Receiver	Sender-to-Receiver	Sender-to-Receiver	Sender-to-Receiver
Time-stamping	App. layer	MAC	MAC	MAC
Sensitive to delays	propagation, decoding and interrupt handling	encoding, decoding and interrupt handling	propagation	propagation
Flooding	No	No	Frequent	Infrequent
Skew estimation	No	No	8LR	2LR
Compensation of propagation delay	No	Yes	No	Yes
Request-Reply	No	Iterative	No	Recursive
Adaptive	No	No	No	Yes
No. of messages (per node)	N (No of nodes)	2	1	0.14 (in long run)
Energy usage	Very high	High	Low	Very Low
Accuracy (per-hop)	29.1 μ s (single-hop)	20 μ s	0.5 μ s	0.23 μ s

4.8 Conclusions

The RTSP algorithm for global time synchronization in WSNs is described, which gives an average accuracy of $0.23\mu\text{s}$ per hop in a large multi-hop clustered network using 7-times lesser energy than that of FTSP in the long run. An analysis of the sources of errors shows that the two sources of errors are variation in propagation delays and relative drift between local clocks, which are duly compensated by the algorithm. The accuracy of RTSP is improved by using the MAC-layer time-stamping based on SFD byte, which is simpler and more accurate. Further improvement in accuracy is gained by the compensation of propagation delay and adjustment of the timestamps at each hop. The RTSP algorithm achieves energy-efficiency by using several techniques, which include the infrequent broadcasts by the reference node, skew-estimation using 2LR instead of 8LR, and reducing the number of time synchronization requests through the adaptive re-synchronization interval and aggregation of synchronization requests.

CHAPTER 5.

COVERAGE, CONNECTIVITY AND COMMUNICATION

The issues of coverage, connectivity and communication are of primary concern in WSNs. As WSNs are usually over deployed, it is useful to keep redundant nodes sleep until some neighboring node fails or exhausts its energy. However, identification of the redundant nodes should be done very carefully so that coverage and connectivity of the network is preserved.

While coverage is the ratio of area sensing-covered by the sensors to the total deployment area of the network, connectivity is the availability of path between each node of the network. Some applications need partial coverage whereas others need k -coverage. Similarly, some applications need simple connectivity while others need k -connected network for fault tolerance. It is important to note that infrastructure creation and maintenance consumes a lot of energy in WSNs. Therefore, energy-efficient algorithms are needed for coverage, connectivity and communication in WSNs.

This chapter proposes an energy-efficient protocol for coverage, connectivity and communication (C3) which ensures partial coverage, connectivity and efficient communication.

5.1 Related Work

This section provides a quick overview of sleep-based methods for coverage and connectivity, and cluster-based methods for communication in WSNs. A detailed description of these methods is given in chapter 2.

The process of configuring and re-configuring of network topology after deployment is known as topology control. Sleep-based topology control techniques involve both connectivity and coverage, which aim at finding which sensor nodes need to be active at a given time. Several techniques have been proposed for sleep-based topology control, which mostly work in a similar way: each node uses some local rules and observations to transition between sleep, test, and active modes. Examples of sleep-based connectivity methods include the Basic Energy-Conserving Algorithm (BECA) [47], Adaptive Fidelity Energy-Conserving Algorithm (AFECA) [47], Geographic Adaptive Fidelity (GAF) [48], Cluster-based Energy Conservation (CEC) algorithm [33], Adaptive Self-Configuring sEnsor Networks Topologies (ASCENT) [49], SPAN protocol [50], etc.

As topology control involves both connectivity and coverage, several techniques are proposed which are almost similar to the previous techniques (i.e., connectivity only)

except that they activate sensor nodes using coverage-based eligibility rules. Examples of sleep-based coverage (and connectivity) methods include the Probing Environment and Adaptive Sleep (PEAS) [51], the sponsored sector [52] approach, the Lightweight Deployment-Aware Scheduling (LDAS) protocol [53], the integrated Coverage and Connectivity Protocol (CCP) [39], the Lightweight Deployment-Aware Scheduling (LDAS) protocol [53], the Optimal Geographical Density Control (OGDC) [219], the Layered Diffusion-based Coverage Control (LDCC) [220], etc.

A detailed description of CCP and LDCC is provided here and their performance is compared with the C3 protocol in the next sections.

The integrated Coverage and Connectivity Protocol (CCP) [39] uses three states: sleep, listen, and active. It uses HELLO, WITHDRAW, and JOIN messages; and also few timers. Sensors in sleep state, after a timer T_s , go to the listen state and remain in that state for T_l . They assess their eligibility (to become active) on receiving HELLO, WITHDRAW, or JOIN message. A node is eligible if its region is K -covered. Ineligible nodes go to sleep state after T_l , whereas the eligible nodes start a join timer T_j , and wait for any JOIN message. It goes to sleep state if any JOIN message is received, or otherwise transmit JOIN message and go to active state. All active nodes periodically transmit HELLO messages to tell their active state. Any active node that receives this HELLO message checks its eligibility (to remain active). If it is ineligible to remain active, it waits for withdraw timer T_w , transmits WITHDRAW message, and goes to sleep state. K -coverage implies K -connectivity if $R_c \geq 2R_s$ [39], otherwise a hybrid

technique merging CCP with SPAN can be used. The CCP scheme also assumes that the sensor nodes are location aware.

The Layered Diffusion-based Coverage Control (LDCC) [220] uses the idea of triangular tessellation. The normal way of applying triangular tessellation is to deploy sensor nodes at optimal positions, i.e., the corners of triangles. However, in many applications using cheap sensors, sensors are usually deployed randomly with higher density or redundancy. In order to minimize the total number of active nodes, nodes at optimal position should be kept active while others should be allowed to sleep. This requires location information that may not be available in some cases. Therefore, LDCC uses hop-count information in the process of triangular tessellation after WSN deployment. In LDCC, a node at hop h broadcasts an active message, and the nodes at hop $h+1$ receiving the first active message ever reset their timer or wait to a random number. The node whose timer expires first becomes active and broadcasts an active message. After receiving two active messages from the same hop neighbors, the receiving node goes to sleep state for a specified time period T . After T , all nodes become active again to run the above steps in the next round. The LDCC protocol attains higher coverage ratio and lower message-overhead than other similar protocols.

In the large multi-hop WSNs, cluster-based routing methods may significantly improve the robustness and energy consumption of the network. There are several cluster-based routing protocols where gateways are used to maintain communication between two or more clusterheads. Some protocols use power metric where mobile

terminals adjust their transmit power depending on the distance between communicating nodes [76], [77]. For example, the Common Power (COMPOW) protocol [78], [79] in which all the nodes transmit at the same power to ensure bidirectional links, the Cluster Power (CLUSTERPOW) protocol [80] that divides the network into virtual clusters with no clusterheads and allows different clusters to have different power levels, and the Warning Energy Aware Clusterhead (WEAC) protocol [73] that divides the network into clusters which are managed by clusterheads and connected by gateways. However, these protocols do not address sleep scheduling techniques. In the Neighbor-Aware Clusterhead (NAC) protocol [81], nodes are sleep-scheduled by clusterheads without affecting the connectivity and coverage of the network. Gateway may choose to have more frequent wakeup times to satisfy the connectivity and coverage issues.

5.2 Basic Principals of C3

The C3 protocol is based on the following ideas.

5.2.1 WSN as a Set of Rings

It is possible to divide the WSN deployment area into virtual rings of required width as shown in Figure 5.1 (sink at center). The optimal width of the rings is determined by R_c , where R_c is the communication range of nodes.

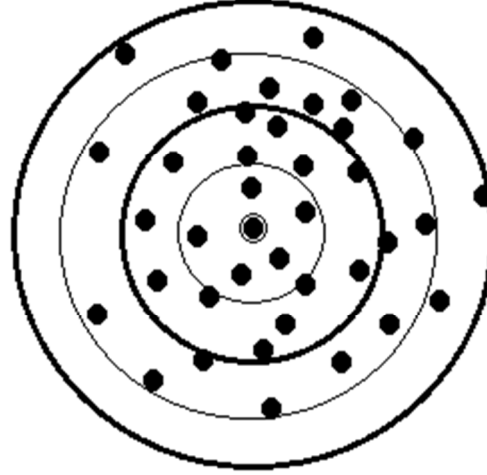


Figure 5.1: WSN as a set of rings (sink is at the center)

The rings of optimal width may help in establishing network connectivity. For example, Y. Zhou and M. Medidi [221] have proposed a multi-parent staggered wakeup scheduling MAC (MS-MAC) for WSNs, which organizes the sensor nodes as a set of rings having an optimal width of $\mu = R_c/2.45$. For a specific node in ring M , the number of potential forwarders in the upper ring $M-1$, i.e., closer to the base station, is determined by the network density ρ (nodes/m²). It assumes that the nodes are location aware and are randomly deployed, and the base station is located at ring 0. However, such a scheme results in a route-length that is more than twice the shortest path which significantly increases the end-to-end delay.

In C3 protocol, the above idea of ring formation is extended. However, C3 does not assume nodes to be location aware; instead it uses the received signal strength indicator (RSSI) for distance estimation in order to define rings in the network. Moreover, the network is divided into clusters where the probability of a node to be

selected as clusterhead is higher in odd-numbered rings during one round but higher in even-numbered rings during the other round. That is odd or even numbered rings preferably form clusterheads in a specific round while neighboring nodes in the lower ring join the cluster, which makes route-length almost equal to the shortest path.

5.2.2 Triangular Tessellation Based Deployment

A static network can be planned in advance and deployed optimally using schemes such as square, hexagon, etc. The triangular tessellation is however considered as the most optimal deployment of sensors in WSNs [220]. Figure 5.2 shows some examples of the optimal deployment geometries.

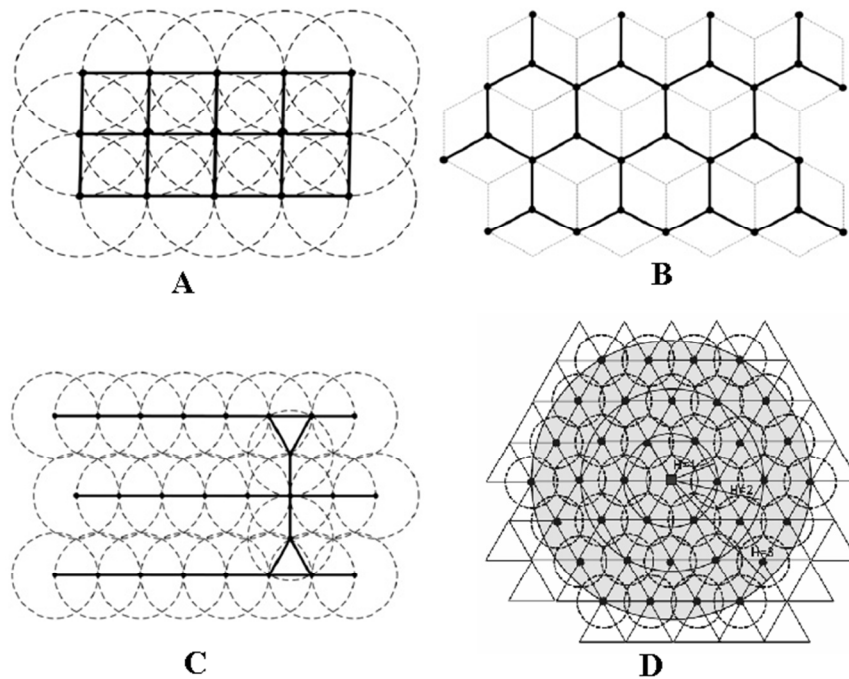


Figure 5.2: Optimal deployment geometries.

(A) Square, (B) Hexagon, (C) Strip, and (D) Triangular Tessellation [220], [222].

The idea of optimal deployment has been exploited even after the deployment in sleep scheduling schemes, for example, where nodes at optimal position are kept active while others are allowed to sleep. For instance, B. Wang et al. [220] have proposed Layered Diffusion-based Coverage Control (LDCC) that uses the idea of triangular tessellation.

It is important to note that in order to provide total connectivity-coverage, the distance between two nodes should be $\min\{\sqrt{3}R_s, R_c\}$ [222], where R_c is the communication range and R_s is the sensing range of nodes.

5.2.3 RSSI-based Distance Estimation

The Received Signal Strength Indicator (RSSI) measures the power of a received signal. The 802.11 standard commonly implements it. The value of RSSI can be used to estimate the distance between two nodes where distance is indirectly proportional to RSSI. As the RSSI value may be increased due to the presence of noise, Link Quality Indicator (LQI) is also provided by some standards. However, RSSI and LQI are perfectly correlated [223] as shown in Figure 5.3, which means that any of the two can be used for distance estimation.

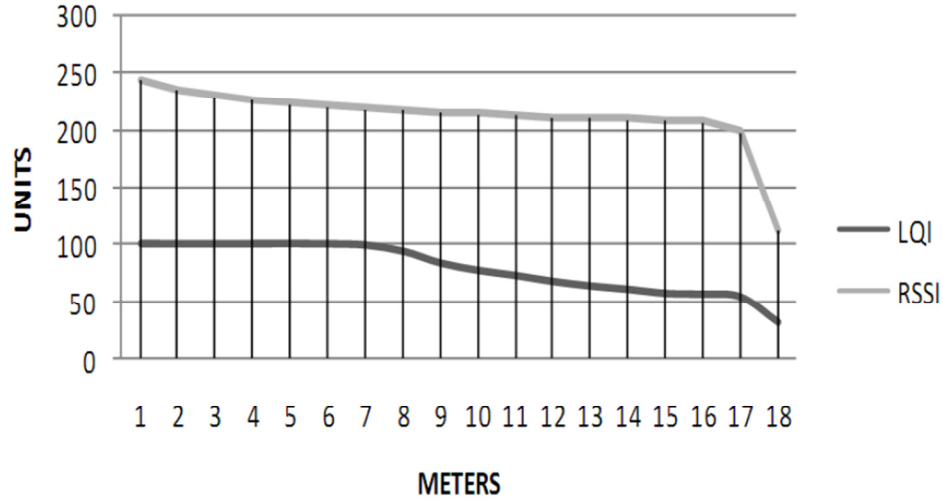


Figure 5.3: Distance vs. LQI and RSSI values [223]

RSSI is used for distance estimation in C3 protocol. This is because RSSI is better indicator of distance especially when distance should be estimated by using only one measurement instead of several measurements that can be averaged out [224].

5.3 System Model for C3 Protocol

First, the assumptions and message structure in C3 protocol are briefly described.

5.3.1 Assumptions

Following assumptions, which are realistic and commonly found in literature, are made in the proposed algorithm:

1. Nodes have unique IDs from 0 to $n-1$.
2. The network-deployment is random, dense and uniform.

3. Controlled flooding of messages is possible.
4. RSSI values are available for the received signals.

5.3.2 Distance Estimation

RSSI is used to estimate distance of the receiver from a sender. The value of RSSI near the sender is highest as shown in the Figure 5.4. Nodes that receive a signal stronger than Threshold1 make the first ring while nodes that receive a weaker signal that is stronger than Threshold2 make the second ring.

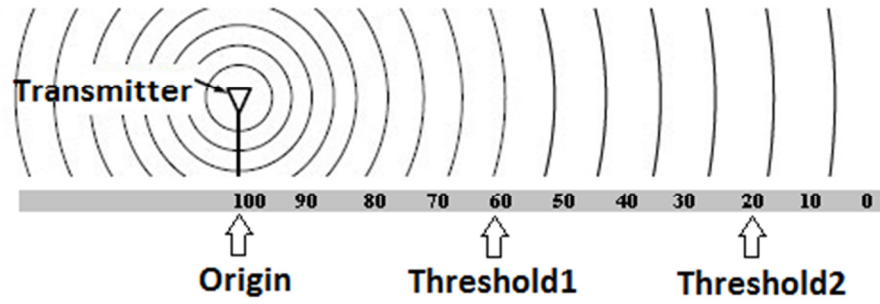


Figure 5.4: RSSI, distance, and ring formation.

5.3.3 Structure of the C3 messages

The C3 algorithm achieves its objectives by exchanging messages in the format that is shown in Table 5.1. Moreover, nodes maintain a few variables, including myID, myCH, myHopNo, myRingNo, myDingNo, myEnergy, myUpCH, myDownCH, myPeerLeftCH, myPeerRightCH, etc.

TABLE 5.1: Structure of a C3 Message

Field	Description
msgType	initHopRing, initCovRing, joinMe, iJoinYou
msgID	ID of the message
senderID	ID of the message sender
destID	ID of the destination node
txPower	Transmission power for the message
roundNo	Current round number
hopNo	Hop number of the sender
ringNo	Ring number of the sender
dingNo	Ring number of the sender node inside a cluster
myStatus	Fully Active; Sensing Only; Sleeping

5.3.4 Node States

Following node states are used by the proposed algorithm:

1. **Fully Active:** A fully active node can sense its environment and can transmit and receive messages. Nodes in this state consume higher energy.

2. **Sensing Only:** A sensing only node can sense its environment but cannot transmit or receive messages. A very low energy is consumed in this state.
3. **Sleeping:** A sleeping node can neither sense its environment nor can it transmit and receive messages. Extremely low energy is consumed in this state.

5.4 The C3 Protocol

Coverage, Connectivity and Communication (C3) runs in four steps as follows:

1. **Formation of Rings:** it divides the network deployment area into virtual rings of optimal width μ .
2. **Formation of Clusters:** it makes clusters such that the nodes at alternating rings have higher probability of being selected as clusterheads.
3. **Formation of Dings:** it ensures partial coverage by applying triangular tessellation inside a cluster. A ding is nothing but a ring inside a cluster.
4. **Routing or Communication:** Finally, it allows nodes to send data through cluster heads which may involve gateways.

These steps are described in algorithmic form the following subsections.

5.4.1 Formation of Rings

The C3 algorithm runs the following steps to define rings:

5. Sink: The sink (or reference node) broadcasts an initHopRing message with roundNo=0, hopNo=0, ringNo=0 at R_C (i.e., normal communication range.)
6. Receiver Nodes: A receiving node finds its hop no. and ring no. based on RSSI as follows:

Algorithm 1: makeRings

1. **If** myHopNo > msg→hopNo **then**
 2. **If** RSSI > Th1 **then**
 3. myRingNo=2*(msg→hopNo)+1;
 4. **Else if** RSSI > Th2 **then**
 5. myRingNo=2*(msg→hopNo)+2;
 6. **Else**
 7. myRingNo=2*(msg→hopNo)+3; // be updated if stronger signal received
 8. myHopNo = msg→hopNo + 1;
 9. **End if**
 10. myHopNo = msg→hopNo + 1,
 11. Broadcast initHopRing message with msg→hopNo=myHopNo,
 msg→ringNo=myRingNo.
 12. **End if**
-
-

Here Th_1 is the RSSI value at a distance from the transmitter which is equal to 40% of the communication range R_c .

5.4.2 Formation of Clusters

Clusterhead: After some wait $w = SD * (1/myEnergy) + ((roundNo + ringNo) \% 2)$, a node announces to become cluster head (CH) by broadcasting a joinMe message as follows:

Algorithm 2: makeClusters

1. $w = SD * (1/myEnergy) + ((roundNo + ringNo) \% 2)$
 // SD is standard deviation of residual energies of neighboring nodes.
 2. Wait (w);
 3. **If** a joinMe message with stronger RSSI from same or upper ring received **then**
 4. Call joinCluster
 5. **Else**
 6. myCH = myID;
 7. Broadcast joinMe message
 8. Set CH to FULLY ACTIVE state
 9. **End if**
-

Non-Clusterheads: A node that receives a joinMe message from a CH (at same or neighboring ringNo) may join it as follows:

Algorithm 3: joinCluster

1. **If** msgType==joinMe AND myRingNo is same or ± 1 as msg→ringNo **then**
 2. **If** (RSSI > Th1 AND myRingNo==msg→ringNo) OR
 3. (RSSI > Th2 AND myRingNo==msg→ringNo+1) **then** // join CH.
 4. myCH = msg→senderID;
 5. Send acknowledgement by iJoinYou message.
 6. **Else if** (RSSI < Th1 AND RSSI > Th2 AND myRingNo==msg→ringNo) OR
 (RSSI < Th2 AND (myRingNo==msg→ringNo OR
myRingNo==msg→ringNo ± 1)) **then**
 //become CH... on the same ring at a reasonable distance
 7. Call makeClusters
 8. **End if**
 9. **End if**
-

Clusterheads wait for some time to receive iJoinYou message from sensor nodes and then the following steps:

Algorithm 4: makeLinks

1. Wait for replies and iJoinYou messages
 2. CHs check their connectivity to other CHs: parent, child and peer rings.
 3. Define gateways, if needed; i.e., find a gateway node, if no direct neighbor is available in the upper hop/ring... if no gateway in upper hop/ring, find at peer.
 4. All other nodes declare themselves as CH and make clusters
 5. CHs and gateways are fully active nodes, while others are eligible to sleep (or sensing only) subject to the coverage-control step.
-

5.4.3 Formation of Dings

Within each cluster, a CH starts the process of ding formation by broadcasting initCovRing message with $\text{msg} \rightarrow \text{dingNo} = 0$ within a cluster. Note that all nodes active and listening in the beginning. All receiving nodes in a cluster do the following:

Algorithm 5: makeDings

1. **If** $R_C \leq \sqrt{3}R_s$ **then**
 2. Return // no need to make dings
 3. **Else**
 4. $\text{dingNo} = 1 + (R_C \% \sqrt{3}R_s)$; // Nodes update their dingNo
 5. **End if**
-

5.4.4 Sleep-scheduling

Starting from nodes with a small dingNo, i.e., a clusterhead sends out initCovRing message with myStatus= FULLY ACTIVE to its neighbours (upon expiration of its timer). The receiving nodes in the cluster set their modes as follows:

Algorithm 6: setModes

1. Check if initCovRing message with some value in myStatus is received;
 2. **If** one message with myStatus as SENSING ONLY or FULLY ACTIVE message received from same dingNo or dingNo+1 **then**
 3. Reset timer;
 4. wait (dingNo);
 5. **Else if** two or more such messages received from neighbors with the same dingNo or dingNo+1 **then**
 6. Set mode to SLEEPING for time period T.
 7. **Else**
 8. Set mode to SENSING ONLY for time period T.
 9. **End if**
-
-

5.4.5 Routing or Communication

The nodes follow the following rules for communication:

Algorithm 7: msgRouting

1. **If** myRingNo ≤ 2 **then**
 2. Send data directly to the sink.
 3. **Else if** an ordinary node **then**
 4. Send data to your CH.
 5. **Else if** it is a CH **then**
 6. **If** a link to CH in upper ring is available **then**
 7. Send data to neighboring CH in the upper ring.
 8. **If** no neighbor CH in upper ring **then**
 9. Send data to a gateway.
 10. **If** no gateway available **then**
 11. Send data to a peer CH.
 12. **End if**
 13. **End if**
 14. **Else if** more than 2 CHs or links are available at a certain level **then**
 15. Pick the one CH randomly or with higher energy
 16. Send data to the selected CH.
 17. **Else if** no link to any neighbor is available **then**
 18. Broadcast at maximum transmission power.
 19. **End if**
 20. **End if**
-

5.5 Analysis of C3 Protocol

In this section, C3 protocol is analyzed in terms of connectivity, coverage, and end-to-end delay.

5.5.1 Connectivity

As C3 protocol uses rings for connectivity, it is important to analyze the network connectivity in terms of ring width. As the nodes in 1st and 2nd ring can directly access the sink, let us consider a case where a node in 3rd ring wants to send data to the sink. The network connectivity depends on the total number of nodes present in the shaded area in Figure 5.5.

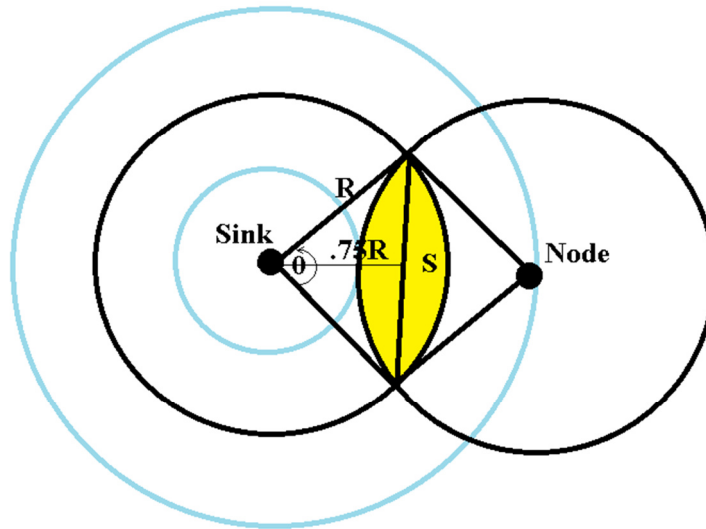


Figure 5.5: The potential forwarders or CHs in the shaded area

The total number of nodes (i.e., clusterheads or gateway nodes) K_N in the shaded area can be calculated by finding the total area of circular segment S, multiplying it by 2, and then multiplying it by the network density ρ that is nodes/meter²:

$$K_N = 2\rho \left[\frac{R^2}{2} (\theta - \sin \theta) \right] \quad (1)$$

$$K_N = \rho R^2 (\theta - \sin \theta) \quad (2)$$

The angel θ can be calculated as follows:

$$\theta = 2\cos^{-1}\left(\frac{0.75R}{R}\right) \quad (3)$$

$$\theta = 1.445 \quad (4)$$

As $\mu = R/2.45$, therefore $R = 2.45\mu$. Putting the value of R and θ in equation (2) gives the following equation:

$$K_N = \rho (2.45\mu)^2 (1.445 - \sin 1.445) \quad (5)$$

Solving equation (5) results in the following equation:

$$K_N = 2.721\rho\mu^2 \quad (6)$$

It is clear from equation (6) that the total number of potential forwarders K_N in the shaded area is determined by network density ρ and width of rings μ where $\mu=R/2.45$. For example, if $R=15$ m (and hence $\mu=6.122$) and $\rho=0.08$, then K_N will be 8. In other words, the total number of potential forwarders in the shaded area depends on the communication range R_c and the network density ρ .

In C3, one or more always-on clusterheads in the shaded area will serve as potential forwarders for the neighboring nodes in an outer ring.

After a WSN is deployed, it is divided into virtual rings of width μ and then into clusters as shown in the Figure 5.6.

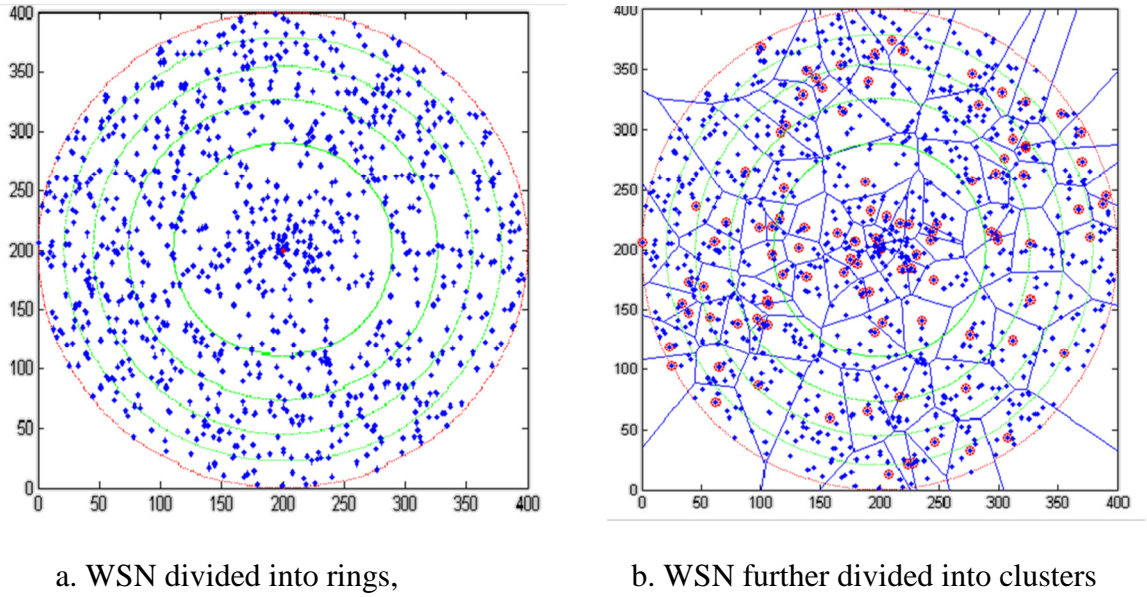
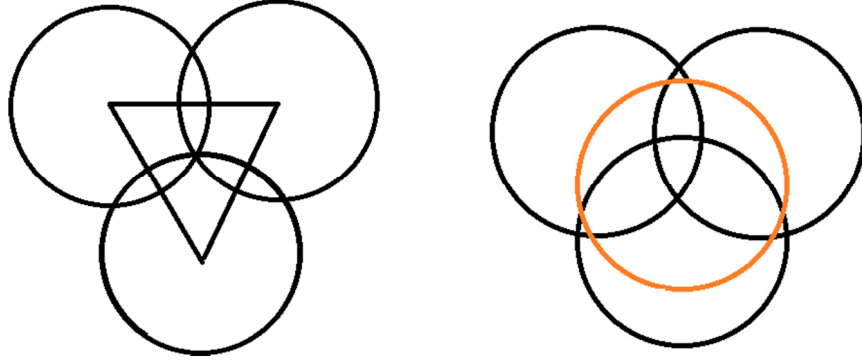


Figure 5.6: WSN divided into rings and clusters by the simulator

5.5.2 Coverage

The C3 protocol uses triangular tessellation for establishing single coverage in the network. It lets nodes on the triangle corners to be actively sensing while putting the redundant nodes to sleep mode. Figure 5.7(a) shows the process of triangular tessellation while Figure 5.7(b) shows a sensing redundant node in the middle.



a. Triangular Tessellation for coverage, b. Redundant node in the middle

Figure 5.7: Triangular Tessellation for optimal coverage

The process of triangular tessellation takes place within the cluster. The cluster head in the central ring (i.e., a ring inside the cluster) starts the process while nodes in the neighboring ring respond as shown in the Figure 5.8.

The width of each side of the triangle is approximately $\sqrt{3}R_s$. However, the achieved coverage is dependent on the availability of nodes at required positions and the accuracy of distance estimation based on RSSI.

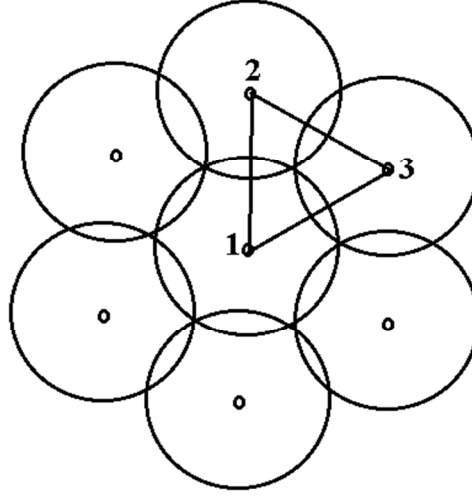


Figure 5.8: Triangular Tessellation for optimal coverage within a cluster

5.5.3 End-to-end Delay

The C3 protocol uses virtual rings like MS-MAC [221]. This analogy can help us analyzing the end-to-end delay in the network. Y. Zhou and M. Medidi [101] have found that the end-to-end delay d can be given by equation (7) as follows:

$$d = 2.45N\Delta + T_{\text{sleep}}/k \quad (7)$$

where N is the number of hops, Δ is the packet transmission delay, and T_{sleep}/k is the staggered wakeup time for k nodes.

However, as C3 protocol keeps at least one clusterhead or a gateway always in active mode, $T_{\text{sleep}} = 0$. This results in equation (8) as follows:

$$d = 2.45N\Delta \quad (8)$$

As the C3 protocol mostly makes clusterheads in the alternating rings and communication is done through clusterheads (may involve gateways if no direct link available), the number of hops N is halved. This gives us equation (9) as follows:

$$d = 1.225N\Delta \quad (9)$$

It is clear from equation (9) that the end-to-end delay in C3 is slightly greater than the delay in fully active network which is $d=N\Delta$. This is because C3 protocol results in a route-length that is slightly longer than the shortest path.

5.6 Performance Evaluation

In order to evaluate the performance of C3 protocol, a simulation is done in MATLAB using the parameters shown in Table 5.2. The simulation was run to collect data on average coverage ratio, packet delivery ratio, number of fully active nodes, lifetime of the system, and energy consumption in the long run.

5.6.1 Energy Model

The radio energy dissipation model by Heinzelman et al. [225] is used for estimating the energy consumed in transmission (E_{Tx}) and reception (E_{Rx}). The energy consumed in transmission (E_{Tx}) of k bits is given by the following equation:

$$E_{Tx}(k, d) = kE_{elec} + kE_{amp}d^\alpha \quad (10)$$

where k is the packet length in bits, d is the distance between transmitter and receiver,

E_{elec} is the electronics energy which depends on many factors including signal's digital coding, modulation, filtering, and spreading, $E_{amp} \in \{E_{fs}, E_{mp}\}$ is the amplifier energy used in free space and multipath respectively depending on the distance and the tolerable bit-error rate, α is the path-loss exponent that is $2 \leq \alpha \leq 4$. The simplified or explanatory form of the equation (10) is given as follows, where threshold distance $d_0 = \sqrt{E_{fs} / E_{mp}}$:

$$E_{Tx}(k, d) = \begin{cases} kE_{elec} + kE_{fs}d^2, & d < d_0 \\ kE_{elec} + kE_{mp}d^4, & d \geq d_0 \end{cases} \quad (11)$$

Similarly, the energy consumed in reception (E_{Rx}) of k bits is given as follows:

$$E_{Rx}(k) = kE_{elec} \quad (12)$$

In other words, E_{Tx} is the sum of electronics energy and amplifier energy, while E_{Rx} is only the amplifier energy. Hence total energy consumed in receiving and forwarding a message is equal to the sum of E_{Tx} and E_{Rx} . It is important to mention that the energy consumed in environmental sensing is not considered because it is usually negligible, constant and always same for any protocol.

5.6.2 Simulation Parameters

The parameters for MATLAB simulations are given in Table 5.2. Notice the use of random topology, random walk mobility model and pause time of zero, which makes the network highly dynamic. Any coverage and connectivity protocol that performs better in such a network is expected to perform even better in a static environment.

TABLE 5.2: Simulation Parameters for C3

Parameter	Description
No. of nodes	50, 100, 200, 500
Prob. of becoming clusterhead	5%
Topology	Random
Deployment area	50m x 50m to 500m x 500m
Mobility model	Random Walk with High Mobility (pause time 0)
No of simulations (results averaged)	1000
Channel	Wireless
MAC	IEEE 802.15.4/ ZigBee
Antenna	Omni
Transmission range and Sensing range	10 - 20 m
E_0 (Initial energy of the node)	2 J
E_{elec} (Electronics energy)	50 nJ/bit
E_{fs} (Free-Space amplifier energy)	10 pJ/bit/m ²
E_{mp} (Multi-Path amplifier energy)	0.0013 pJ/bit/m ⁴
E_{DA} (Data-Aggregation energy)	5 nJ/bit/signal

5.6.3 Results and Discussions

The performance of C3 protocol is compared with the CCP and LDCC protocols in terms of average coverage, packet delivery, number of active or clusterhead nodes, communication and coverage lifetime of the network, and energy consumption.

The coverage ratio is defined as the ratio of area sensing-covered by all nodes to the total area of network. Figure 5.9 shows the average coverage ratio of different protocols. The CCP protocol provides 100% coverage in a densely deployed network for any value of R_c/R_s because it activates larger number of sensor nodes which is not optimal. The increase in R_c/R_s results in decreasing the average coverage ratios for both LDCC and C3 protocols. However, average coverage ratio of the C3 protocol is much higher than that of LDCC.

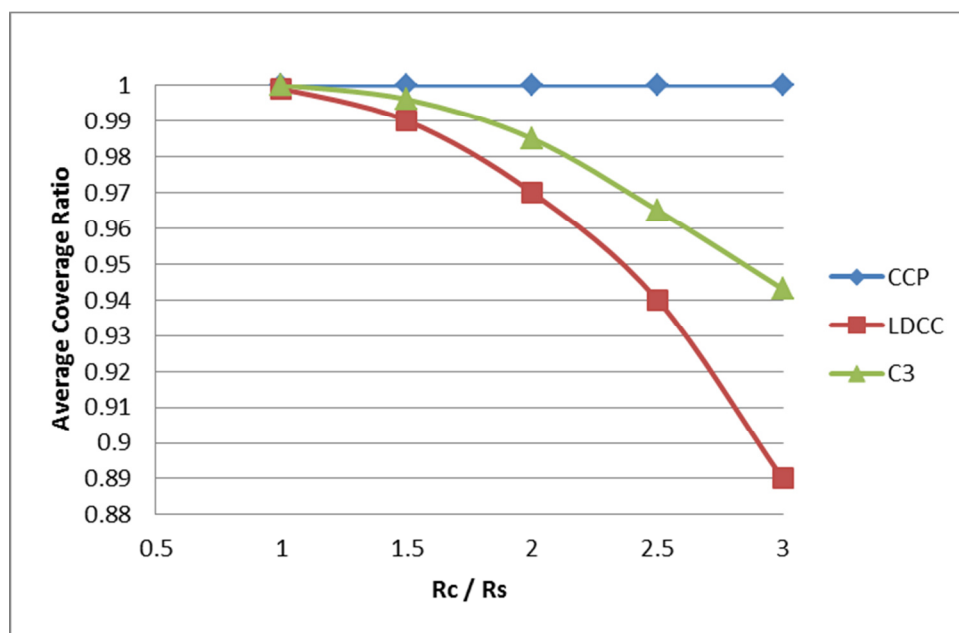


Figure 5.9: R_c/R_s vs. Average Coverage Ratio

A comparison of the packet delivery ratio of different protocols is shown in Figure 5.10. With the increase in communication range (and hence the value of R_c/R_s), packet delivery ratio also increases due to better network connectivity for all the three protocols [39]. However, C3 protocol performs better when $R_c/R_s < 2$ due to the following reasons: 1) nodes communicate through clusterheads and gateways which establish a shorter route to the sink, and 2) it makes sure that a path to the sink exists in a connected network.

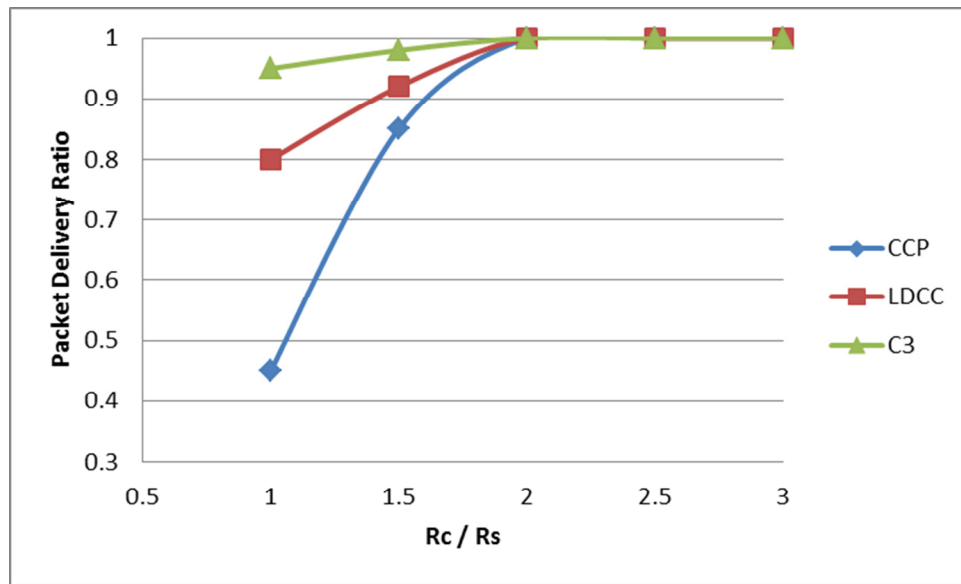


Figure 5.10: R_c/R_s vs. Packet Delivery Ratio

Figure 5.11 compares the total number of active nodes or clusterheads against the value of R_c/R_s . Accordingly, the total number of active nodes or clusterheads decreases when the value of R_c/R_s increases. This is because increase in communication range (R_c) increases the network connectivity [39].

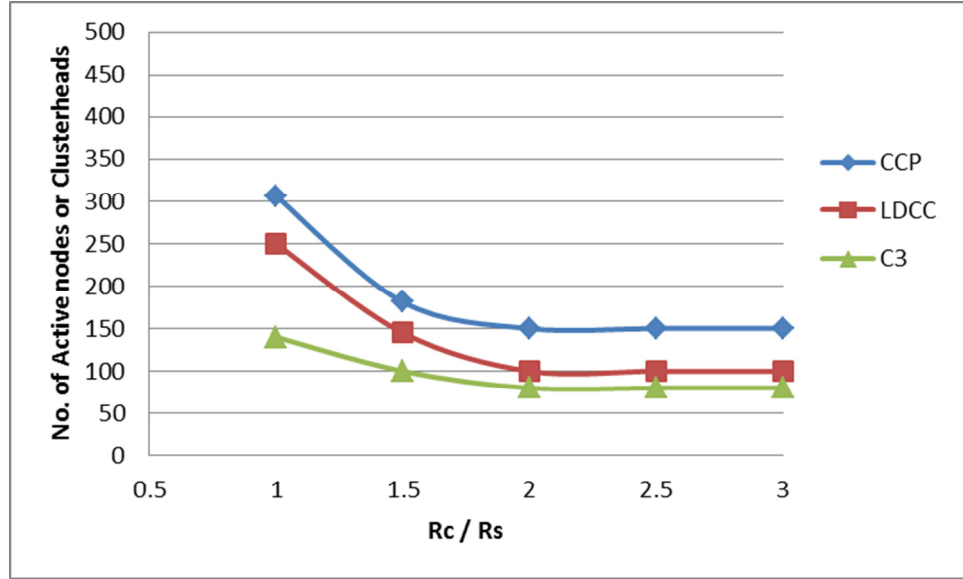


Figure 5.11: Rc/Rs vs. No. of Active nodes or Clusterheads (for n=500)

The lifetime of the system is determined by the time or total number of rounds for which it has some nodes alive. Figure 5.12 shows the total number of nodes that are alive with the passage of time for each protocol. Initially, all the nodes are alive. With the passage of time, nodes start depleting their energy and hence die. After 1000 rounds, all the nodes in CCP deplete their energy, while 24% of the nodes in LDCC and 78% of the nodes in C3 remain alive. A similar trend is seen in the packet delivery ratio which drops due to the decrease in number of nodes alive with the passage of time.

Similarly, Figure 5.13 shows the coverage ratio of network with the passage of time for each protocol. Initially, all the nodes are alive and a coverage ratio of 1 is achievable. With the passage of time, nodes start depleting their energy and hence coverage is reduced. After 1000 rounds, all the nodes in CCP die resulting in no coverage at all, while LDCC still provides 45% coverage and C3 provides 90% coverage.

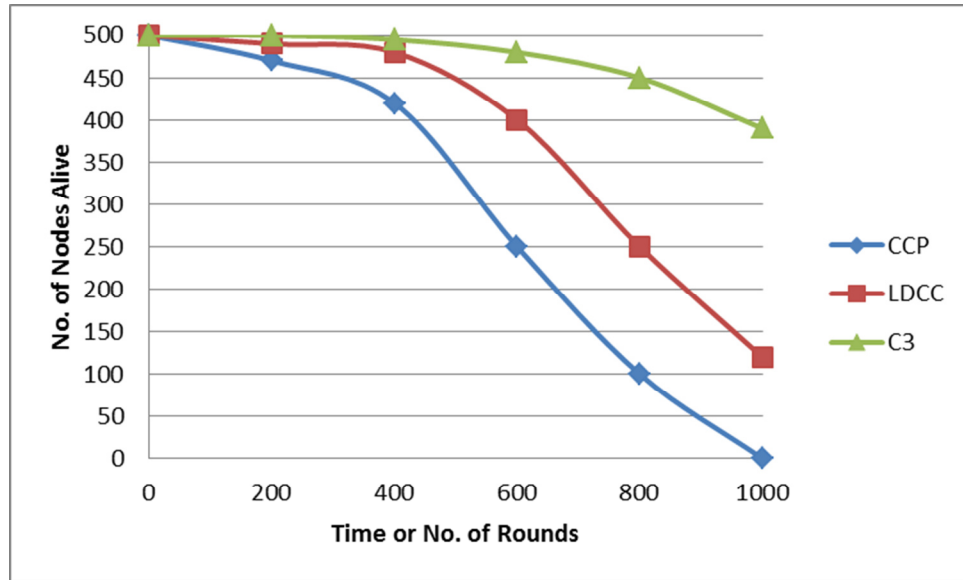


Figure 5.12: Communication Life Time of the System

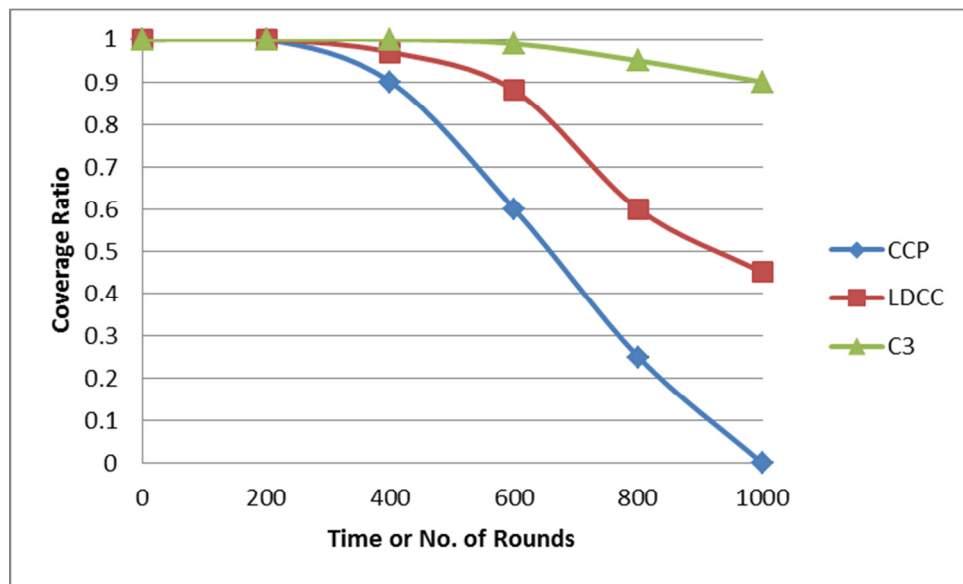


Figure 5.13: Coverage Life Time of the System

The energy consumption of the system is compared in Figure 5.14. Due to the larger number of active nodes (see Figure 5.11) and lower packet delivery ratio (see

Figure 5.10) requiring retransmissions, the CCP and LDCC protocols consume much higher energy than C3 protocol. Consequently, CCP consumes all the sensors' energy after 1000 rounds, while LDCC consumes 76% of the energy and C3 consumes only 22% of the energy.

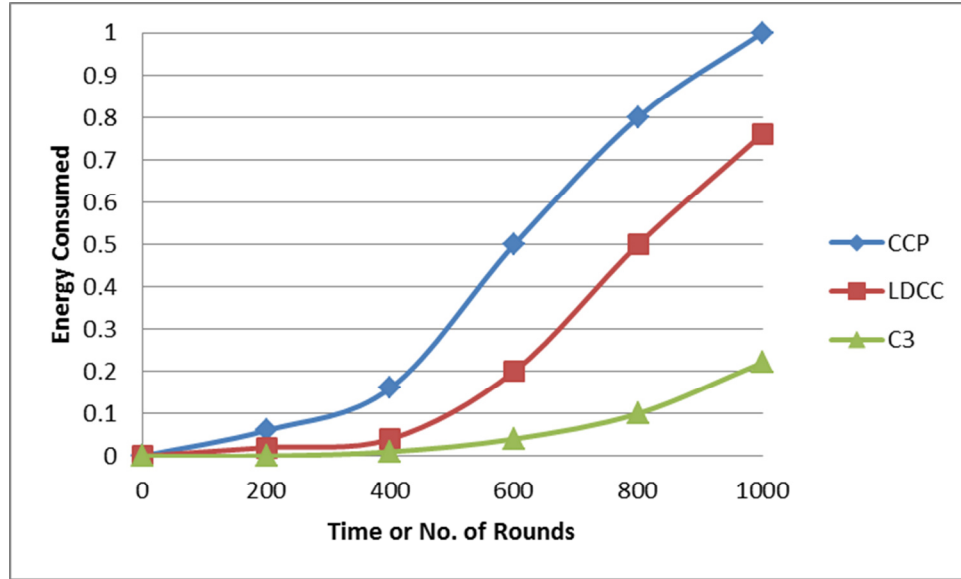


Figure 5.14: No. of Nodes vs. Energy Consumption

The results of simulation show that C3 protocol is better than CCP and LDCC in terms of packet delivery, number of active or clusterhead nodes, communication and coverage lifetime of the network, and energy consumption. However, in terms of average coverage, C3 does not perform better than CCP which provide guaranteed coverage if sufficient redundancy is available. On the other side, C3 provides partial coverage in addition to the guaranteed connectivity and efficient communication. It is suitable for applications such as environment monitoring which require partial coverage and low energy consumption.

5.7 Conclusions

A protocol for coverage, connectivity and communication (C3) in WSNs is proposed. It ensures partial coverage, 1-connectivity and energy efficient communication. The C3 protocol runs in four steps: make rings, define clusters, make dings, and start sending data. First, it divides the network deployment area into virtual rings of optimal width μ . Second, it makes clusters such that the nodes at alternating rings have higher probability of being selected as clusterheads. Third, it ensures partial coverage by applying triangular tessellation inside a cluster. Finally, it allows nodes to send data through cluster heads which may involve gateways. The proposed protocol has better performance when compared to other protocols such as CCP and LDCC. The C3 protocol gives at least 1-connected network depending on the network density and communication range of the nodes. It provides greater than 90% coverage which depends on the availability of nodes at required positions and the accuracy of distance estimation based on RSSI. However, it has end-to-end delay of $1.225N$ which is only slightly greater than the delay in fully active network because of the route-length that is slightly longer than the shortest path.

CHAPTER 6.

SDS DETECTION SYSTEM

Sand and dust storms (SDSs) offer very serious hazards to the environment, economy and health. An early warning of the upcoming SDS would allow people to take precautionary measures. Traditionally, satellite imaging is used to detect large-scale and long-term SDSs. However, small-scale and short-term SDSs may go undetected due to the poor spatial and temporal resolution of satellites. A hybrid design of sand and dust storm detection system (SDSDS) is proposed which uses wireless sensor network (WSN) and satellite imaging in order to detect SDSs of all types. A layered architecture of context-aware systems is used. While the WSN provides real time data from the area of interest, near-real time METEOSAT MSG images are obtained from their website. Such a hybrid approach can effectively detect and predict SDSs of all types.

6.1 Introduction

Sand and dust storms (SDSs) offer very serious hazards to the environment, economy and health, and appear frequently in the Middle-East, Australia, North-Africa and North-China during spring, winter and early summer [85]. There are four main types of SDSs: small-scale SDS (covers a small geographic area), large-scale SDS (covers a

vast geographic area), short-term SDS (exists for a few minutes or hours), and long-term SDS (exists for many days or months) [92].

An early warning of the upcoming SDS can help in avoiding serious consequences by letting people take some preventive measures. However, this requires continuous monitoring of sand and dust in the area of interest. Several technologies are available for sand and dust monitoring, including video-surveillance, sensory information, satellite imagery, unmanned aerial vehicle, etc. However, a particular technology is suitable only for the specific types of SDS. For example, satellite imaging can detect only large-scale and long-term SDSs due to poor spatial and temporal resolution of the images, while WSNs can detect SDSs of all types except large-scale SDSs due to their deployment in a limited area. Therefore, a hybrid approach consisting of WSN and satellite imaging can detect SDSs of all the four types.

A WSN is deployed in a small area of the Arabian Desert to get accurate, firsthand, real-time and continuous data from the source of SDS. Near-real time images from Meteosat RGB Composites Dust for Eastern Africa [226] are used to detect any large-scale SDS in the areas closer to the WSN deployment. Such a hybrid approach improves the system performance in terms of accuracy, energy consumption and cost. A hybrid design is proposed for sand and dust storm detection system (SDSDS) using WSN and satellite imaging in order to detect SDSs of all types.

The rest of this chapter is organized as follows. Section 6.2 briefly describes the related work. Section 6.3 discusses the main issues in SDS detection using WSN and

satellite imaging. Section 6.4 gives the system architecture of SDSDS. Section 6.5 evaluates the performance of experimental prototype. The last section concludes the chapter.

6.2 Related Work

Satellite imaging has been commonly used to detect SDSs since 1970s. The most commonly used satellite imagers for SDS detection are GOES (Geostationary Operational Environmental Satellites), AVHRR (Advanced Very High Resolution Radiometer), MODIS (MODerate Resolution Imaging Spectroradiometer) and Meteosat SEVIRI-MSG.

Mostly, these satellites provide multispectral images which use multiple bands including the optical/visible or red-green-blue (RGB), near-infrared (NIR), middle infrared (MIR) and far infrared (FIR) or thermal, and microwave or radar. While SDSs can be easily detected from ordinary digital images using artificial intelligence (AI) and digital image processing (DIP) techniques, multispectral images need sophisticated techniques such as aerosol optical depth, brightness temperature (BT), thermal properties, refractive index, etc. [97–99].

Although, both visible and near-infrared (VIR) images can be used to detect SDS during daytime, thermal infrared (TIR) images can perform better both at daytime and nighttime using BT-based techniques. For example, thermal properties have been used to distinguish between dust, background and clouds [97–99], [180]; infrared emissivity have

been used to detect sand and dust storms [187], [188]; and refractive index and particle size of aerosol have been used to detect SDS using MODIS thermal channels [189]. Many other studies have used MODIS thermal channels [98], [192], [193]. Meteosat MSG (Meteosat Second Generation) satellite has been used in many recent studies [187], [188]. Meteosat MSG is an European geostationary satellite which provides 96 images daily. It is gaining popularity due to its higher temporal resolution of 15 minutes and unique services. For example, Meteosat RGB Composites Dust images clearly identify dust storms with higher red fraction in the imagery both at daytime and nighttime.

WSNs have been widely used to monitor natural hazards such as flood, earthquake, snow, fires, etc. However, the use of WSNs in SDS detection is new. This technique is becoming very common because wireless sensors deployed in the area of interest can collect and provide real-time and accurate data which can be used to detect and predict the creation of an SDS right from its origin. A WSN-based SDS detection system requires data on atmospheric pressure, surface temperature, humidity, wind velocity and soil moisture [102]. However, strong wind velocity is a necessary, but not sufficient, condition for the creation of an SDS [91], [103]. The threshold velocity of wind varies from region to region and is assumed to be 6.5 m/s in many dust models for small-scale SDS, and 10 to 14 m/s for strong and large-scale SDS [90][127].

Hybrid approaches are gaining attention due to their efficiency and effectiveness. For example, a combination of satellite imaging and video surveillance [127] is used to detect SDS even when there are clouds blocking the satellite view; a combination of

satellite imaging and lidar [101] is used to detect severe SDS over Greece; and a combination of GIS (geographic information system) and WSNs is used to make pollution maps in urban areas [203].

The proposed system, i.e., SDSDS, uses WSN and near-real time images from Meteosat RGB Composites Dust for Eastern Africa [226]. Meteosat calculates the RGB Composite by using the difference of channels as: $R = IR12.0 - IR10.8$, $G = IR10.8 - IR8.7$, and $B = IR10.8$.

6.3 Challenges

Using WSNs and satellite imaging for SDS detection may offer a number of challenges to the designers. A brief description of the main issues is given below:

1. ***Poor resolution of satellite images***: Satellite images have very poor spatial and temporal resolution. For example, Meteosat RGB Composites for Dust are provided every after one hour, and have pixel resolution of 1419x1193. Therefore, satellite imaging can be used only for large scale and long-term SDSs.
2. ***Signal attenuation during SDS***: During the dust storm, when dust particles are in the air, signal attenuation becomes very high which disturbs the wireless communication. However, attenuation prediction models can duly solve this issue [133], [167].

3. ***Sensor burial during SDS***: During SDS, many of the sensors in WSN may get buried under sand or dust, which results in lower connectivity. However, exploiting the multiple available channels may significantly enhance the connectivity [102].
4. ***Context fusion***: Context-aware systems need to resolve any conflicts and fuse the context data from multiple sources. However, data from WSN and satellite images may support each other in detecting SDS of any type.
5. ***Video streaming over WSN***: Due to bandwidth limitations of WSNs, very efficient video compression and streaming techniques are required. The two recent video standards, MPEG-4 and H.264, are strong candidate for this application. Their performance needs to be evaluated before making a choice.

6.4 System Architecture of SDSDS

The proposed system, i.e., sand and dust storm detection system (SDSDS), uses layered architecture consisting of physical layer, middleware (data, semantic and inference, and management) and application layer [227] as shown in Figure 6.1. The system model of SDSDS is shown in Figure 6.2.

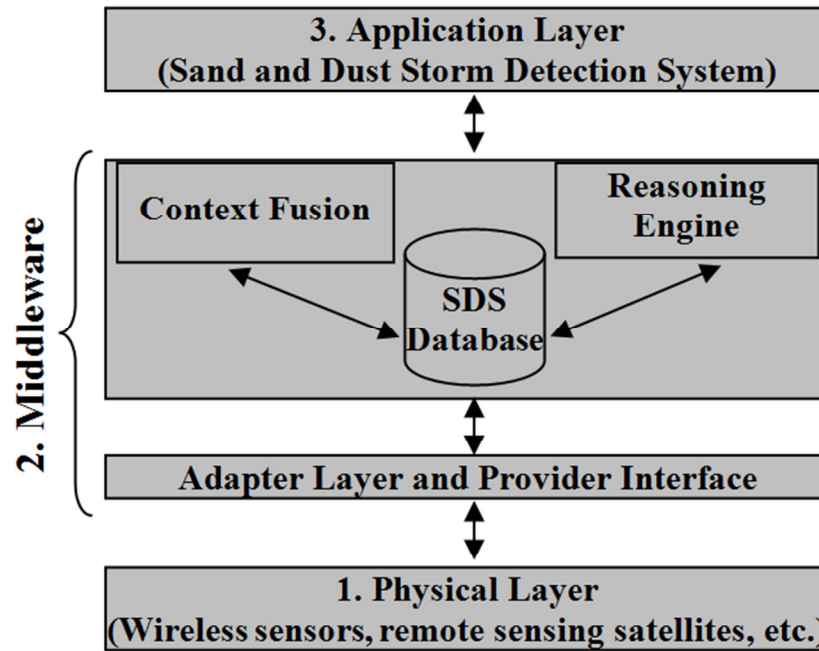


Figure 6.1: Architecture of the proposed system (SDSDS)

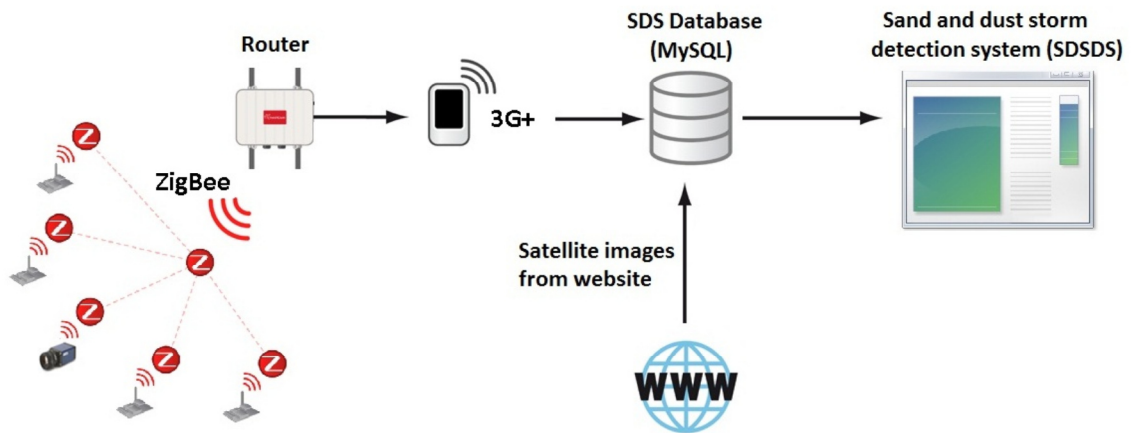


Figure 6.2: System Model of SDSDS

The proposed system (SDSDS) consists of the following main components:

6.4.1 Physical Layer

The Physical Layer of SDSDS consists of wireless sensors deployed in the desert and remote sensing satellites providing the satellite images.

Meteosat MSG satellite provides RGB Composites Dust images every after one hour. The WSN consists of motes which house the sensors for GPS, atmospheric pressure, surface temperature, humidity, wind velocity and soil moisture.

Each sensor mote also houses a gateway that connects wirelessly to a router. The router sends the sensor data to a database server in the control room using 3G+ technology as shown in Figure 6.2. SDSDS uses this database to detect any SDS, and then issues an early warning accordingly as shown in Table 6.1.

6.4.2 Adapter Layer

The Adapter Layer works like a client wrapper for different SDS technologies. It serves as a device driver that allows different SDS technologies to work with the middleware seamlessly.

6.4.3 SDS Database

The SDS Database is a MySQL database which contains the data provided by WSN and satellite imaging. The WSN provide data on sensor ID, timestamp, sensor

location, atmospheric pressure, surface temperature, humidity, wind velocity, and soil moisture. The satellite images are near-real time Meteosat RGB Composites Dust images taken from their website every after one hour [226].

6.4.4 Context Fusion

The middleware is responsible for a number of functions, including the following:

1. Retrieval and processing of raw data from WSN and satellite images.
2. Preprocessing of the data in order to resolve any conflicts, reject outliers, and make inference.
3. Storing, sharing, distributing and publishing the SDS data to different systems, services or applications.

6.4.5 Reasoning Engine

The Reasoning Engine infers the existence of any SDS in the area of interest using both WSN data and images. It first uses the recently taken Meteosat image to detect any large-scale SDS in the close by area and stores the results. It then uses several recent results to detect any long-term SDS in the area. It is very easy to determine SDS from these images using DIP technique because dust is indicated by higher red fraction in the RGB images. Finally, it uses WSN data in order to infer any short-term or small-scale SDS in the area. Based on the finding, SDSDS issues an SDS warning as follows: 1) a severe warning is issued if SDS is detected by both WSN and satellite images, 2) no

warning is issued if SDS is not detected by any of the two, 3) a warning of short-term and small-scale SDS is issued when a dust even is detected by WSN only, and 4) a warning of long-term and large-scale SDS is issued when a dust even is detected by satellite images only.

6.5 Performance of SDSDS

The proposed system is currently an experimental prototype which is deployed on a small-scale. It has shown very good performance in terms of energy consumption, challenge handling, and efficiency and effectiveness.

6.5.1 Energy Conservation

The system uses a very simple technique in order to save energy. As the strong wind velocity is necessary (but not sufficient) condition for SDS, only wind velocity sensor is kept active all the time. All other sensors on the board are kept in sleeping mode. Whenever, the wind velocity reaches above a threshold value, all other sensors are put into active mode.

It is important to note that most of the sensor energy is consumed in transmitting, receiving and overhearing messages. In order to save more energy, the same necessary condition can be used to put the transceiver in sleep mode in order to avoid overhearing and unnecessary transmissions and receptions.

6.5.2 Efficiency and Effectiveness

The proposed system is highly effective and efficient. For satellite images, the system finds the ratio of red component for each pixel, where a higher red ratio indicates dust. This is very simple, fast and accurate method.

The data from WSN is stored in MySQL database. Simple SQL queries are used to manipulate the data. A higher value in the wind velocity column is used as potential indicator of a dust storm in the region of interest, while other parameters are used to confirm it.

6.5.3 Challenge Handling

The challenges mentioned in previous section are properly handled by the proposed system as follows:

1. The poor resolution of satellite images is compensated by using the images from Meteosat RGB Composites Dust for Eastern Africa [226], which are preprocessed for RGB components. The system is required to apply very simple DIP technique to find the ratio of red component for each pixel. A higher red ratio indicates the presence of dust as shown by 1s in Figure 6.3.

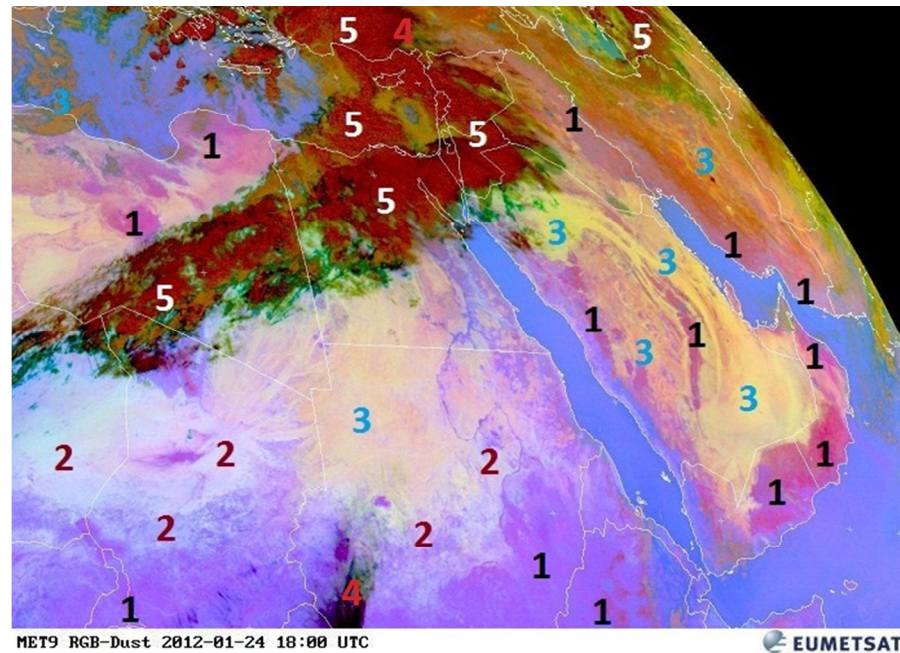


Figure 6.3: RGB for Monitoring of Dust Storms for Daytime & Nighttime.

(1 = dust storm, 2 = clear ground, 3 = low-level clouds, 4 = thin high-level clouds, and 5 = thick high-level clouds)

2. Sensor burial during SDS is avoided by installing them few meters high on the poles. This height increases their communication range which can be helpful during SDS when signal attenuation increases.
3. Context fusion is done by the middleware. It resolves any conflicts and fuses the information from multiple sources, i.e., WSN and satellite. After fusion, data from different sources actually support each other in detecting SDS of all the types.
4. Video compression is done by H.264 which is more efficient than MPEG-4. The video streaming is done through UDP packets of 1KB. The compressed video

generated by the H.264 codec is smoother without fluctuations, smaller in size that is good for low-bandwidth networks, and has less end-to-end delay that will result in efficient transmission over the network.

6.6 Conclusions

Several technologies are available for dust monitoring. However, a particular technology is suitable only for detecting certain types of SDSs. A sand and dust storm detection system (SDSDS) using WSN and satellite imaging is proposed. The WSN is deployed in a small area of Arabian Desert to get real time data from the origin of dust, while near-real time satellite images from Meteosat RGB Composites Dust are taken from their website. The proposed system can efficiently detect SDSs of all types. It achieves energy efficiency by exploiting the necessary condition for SDS, i.e., a strong wind velocity, thereby transceiver and sensors on the node except wind velocity sensor are kept in sleeping mode until the wind velocity reaches above a threshold value. The proposed system is highly effective and efficient due to wireless sensors which provide up-to-date and accurate data from the field and satellite imaging which needs very simple image processing to find the ratio of red component for each pixel, where a higher red ratio indicates the dust. The proposed system achieves higher accuracy due to the middleware that ensures context fusion and conflict resolution, and makes WSN and satellite imaging support each other in detecting SDSs of all types.

CHAPTER 7.

IMPLEMENTATION AND EXPERIMENTS

An experimental prototype is established in order to implement and experiment with the proposed algorithms including the Recursive Time Synchronization Protocol (RTSP), the integrated protocol for Coverage, Connectivity and Communication (C3), and Sand and Dust Storm Detection System (SDSDS). This chapter describes the experimental prototype, implementation and experimentation of algorithms, and the results obtained from these experiments.

7.1 Experimental Setup

A multi-hop network of sensor nodes is deployed in the field. The nodes consist of Crossbow Telosb motes which use the IEEE 802.15.4/ZigBee compliant RF transceiver, have an outdoor range of 75 m to 100 m, provide a data rate of about 250 kbps, have 8 MHz Texas Instruments MSP430 microcontroller with 10kB RAM and 1MB external flash, use CC2420 radio chip, have 32kHz crystal timer, run on TinyOS, and incorporate light, temperature and humidity sensors (TPR2420).

A total of thirteen sensor nodes, including the sink, are pseudo-randomly deployed in an open area as shown in Figure 7.1. Each sensor node is put in the middle of 3D-weaving of 3 sticks to keep the node about 1/3 m above the surface.

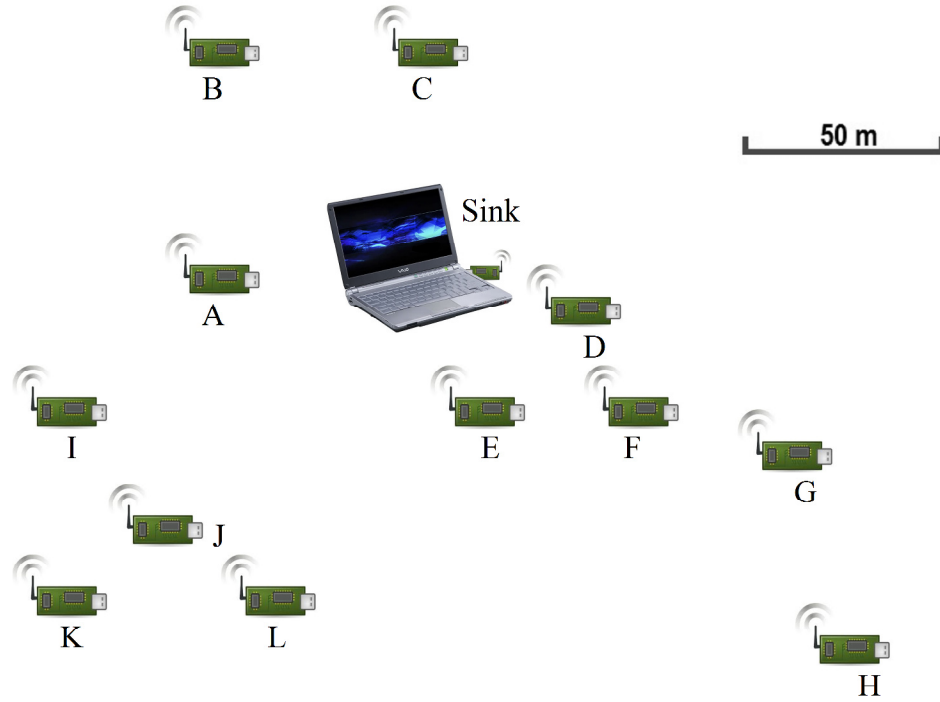


Figure 7.1: Experimental setup.

The experimental prototype uses a Java application interface developed by T. R. Sheltami et al. [228], named as TinyCatcher, which is used to read messages in the network. TinyCatcher actually depends on SerialForwarder that comes with TinyOS and works as a proxy for reading and writing packets from and to the network. Therefore, instead of reading data directly from the serial port, TinyCatcher connects to SerialForwarder. The GUI interface of TinyCatcher and SerialForwarder is shown in Figure 7.2 and Figure 7.3 respectively.

TinyCatcher needs to set two parameters which include 'Address' of the PC running SerialForwarder, and 'Port' used by SerialForwarder. After this, one can connect and listen to the messages or close the connection when finished. Some of the useful functions of TinyCatcher include: 'Get Node Stat' for displaying the data sensed by a particular node, 'Configure Power' for setting the transmission power level of a particular node, 'Add/Remove Node' for adding or removing any node, 'Send Echo' for pinging a node, and 'Discovery' for finding a new or failed node.

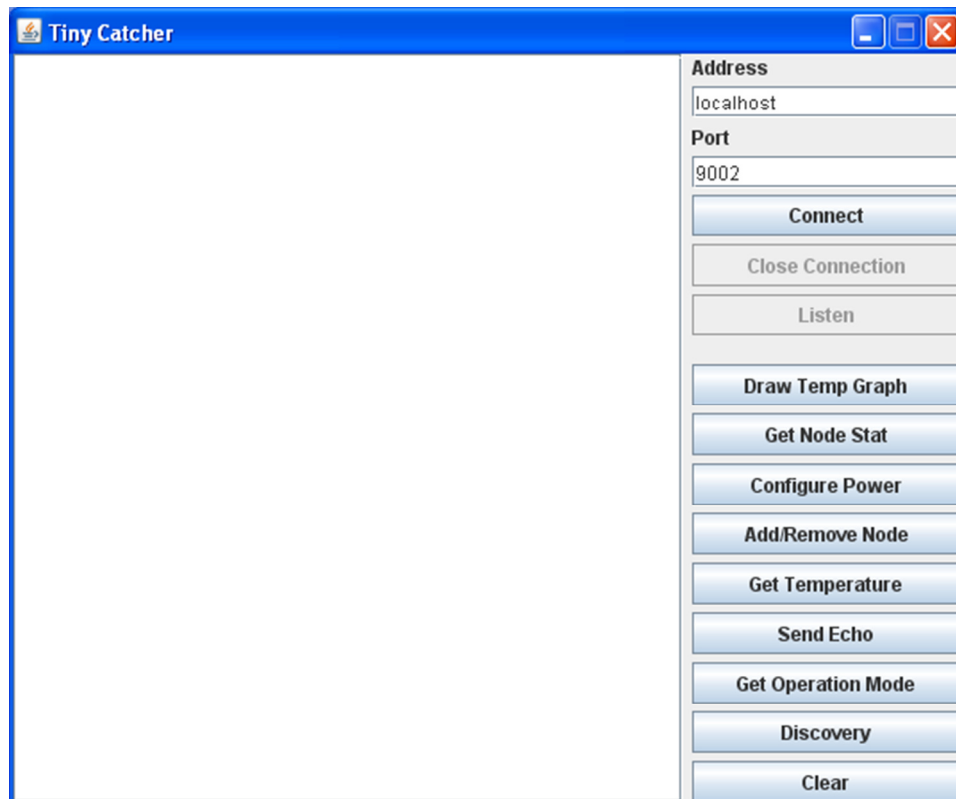


Figure 7.2: TinyCatcher interface.

SerialForwarder needs to set two parameters including 'Server Port' to be opened for server thread, and 'Mote Communication' to specify port number and baud rate as

serial@(serial port):(baud rate). It also displays information on number of packets read, number of packets sent and number of clients connected.

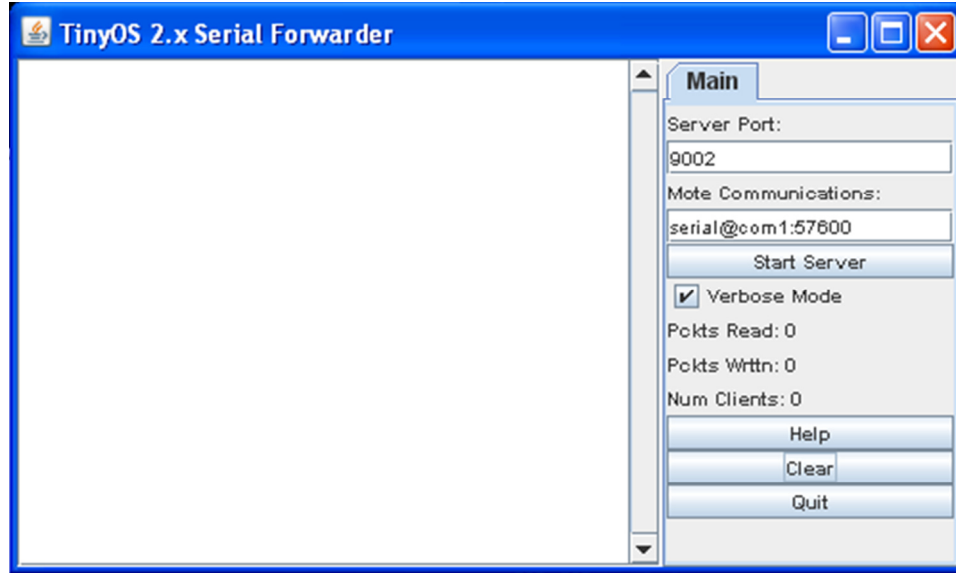


Figure 7.3: SerialForwarder interface.

Using the functions provided by TinyCatcher based on SerialForwarder, several experiments were conducted and very useful results were obtained which are reported in the next few sections.

7.2 Experiments

Several experiments were conducted in order to test the main ideas and principals used by the proposed algorithms, such as clock synchronization using message exchange, distance estimation using RSSI, effective coverage area by sensor nodes, throughput and average delay of the network, and video streaming over the network.

7.2.1 Clock Synchronization using Message Exchange

The implementation of clock synchronization algorithms, mainly FTSP and RTSP, requires a basic function of sending and receiving messages containing the timing information. Based on the information in these messages, nodes can synchronize their clocks with one reference node. To this end, a nesC program was written to get timing information and then send it to other nodes. A test was conducted on two nodes. Figure 7.4 (a and b) shows time on each node after startup. Nodes then wait for a message with timing information to be received. On receiving a message, they set their clocks to the timestamp included in the message, as shown in Figure 7.4 (b and c).

```
Node 1: Started...
Time = 0h 0m 1s
Time = 0h 0m 2s
Time = 0h 0m 3s
Time = 0h 0m 4s
Time = 0h 0m 5s
Time = 0h 0m 6s
Time = 0h 0m 7s
Time = 0h 0m 8s
Time = 0h 0m 9s
Time = 0h 0m 10s
Time = 0h 0m 11s
Time = 0h 0m 12s
Time = 0h 0m 13s
Time = 0h 0m 14s
Time = 0h 0m 15s
Time = 0h 0m 16s
Time = 0h 0m 17s
Time = 0h 0m 18s
Time = 0h 0m 19s
Time = 0h 0m 20s
Guiii Guiii
```

(a) 1st node started

```
Node 2: Started...
Time = 0h 0m 1s
Time = 0h 0m 2s
Time = 0h 0m 3s
Time = 0h 0m 4s
Time = 0h 0m 5s
Time = 0h 0m 6s
Time = 0h 0m 7s
Time = 0h 0m 8s
Time = 0h 0m 9s
Time = 0h 0m 10s
Time = 0h 0m 11s
Time = 0h 0m 12s
Time = 0h 0m 13s
Time = 0h 0m 14s
Time = 0h 0m 15s
Time = 0h 0m 16s
Time = 0h 0m 17s
Time = 0h 0m 18s
Time = 0h 0m 19s
Time = 0h 0m 20s
Time = 0h 0m 21s
Time = 0h 0m 22s
Time = 0h 0m 23s
```

(b) 2nd node started

```
Broadcast beacon <-1> received from base station!
New time after synch
Node 1 [DISC]: Sending DiscoveryMsg<0> with my rank = 11, @Time:3
Time = 0h 0m 3s
```

(c) 1st node synchronized

```
Broadcast beacon <-1> received from base station!
New time after synch
Node 2 [DISC]: Sending DiscoveryMsg<0> with my rank = 2, @Time:3
Time = 0h 0m 3s
```

(d) 2nd node synchronized

Figure 7.4: Clock Synchronization for two nodes.

7.2.2 Distance Estimation using RSSI

The idea of distance estimation using RSSI is used by C3 protocol. To test if it is a good idea to use RSSI for distance estimation, RSSI values were measured at different distances and power levels. Figure 7.5 shows RSSI values in dBm against the distance in meters for three power levels: 1, 16, and 31.

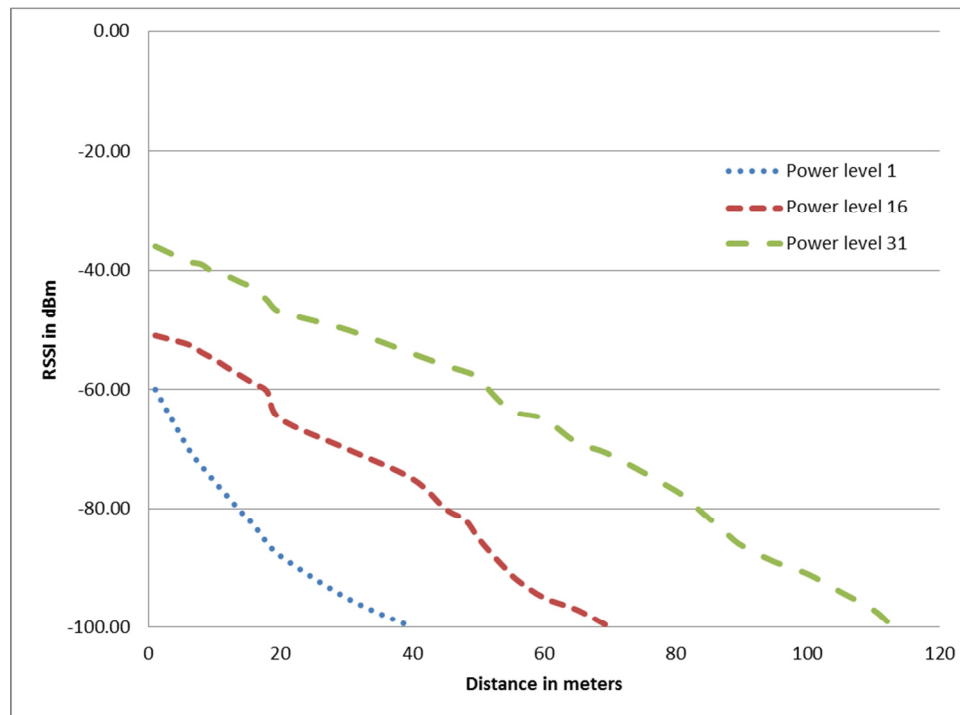


Figure 7.5: RSSI vs. distance.

RSSI measurements range from 0 dBm (strongest) to -100 dBm (weakest), while power levels of a Telosb mote range from 1 (weakest) to 31 (strongest). The results show a very strong correlation between RSSI and distance. Therefore, RSSI is a strong indicator of distance and can be used in C3 algorithm for distance estimation.

7.2.3 Coverage Area

An experiment was conducted to realize the effective area covered by a sensor node. RSSI values were measured in different directions at a distance of 3 meters from the transceiver and plotted as shown in Figure 7.6, where sender node is at the intersection of the two axis of the shape and its USB interface is facing towards the East.

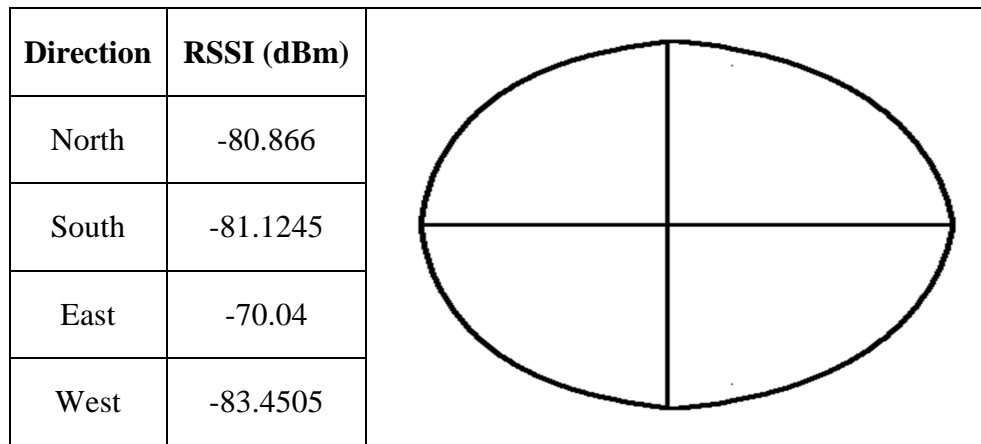


Figure 7.6: Coverage area.

It can be concluded that the coverage area of a node may not be a perfect circle because RSSI values in different directions of the sender may differ due to the environmental differences, transceiver orientation, etc.

7.2.4 Connectivity and Communication

Two experiments were conducted to study the connectivity and communication in the network. The main indicators were the delay and throughput of the network.

Figure 7.7 shows that throughput decreases slightly with the increase in number of hops, but decreases significantly when the number of packets transmitted cross the threshold of 76 packets.

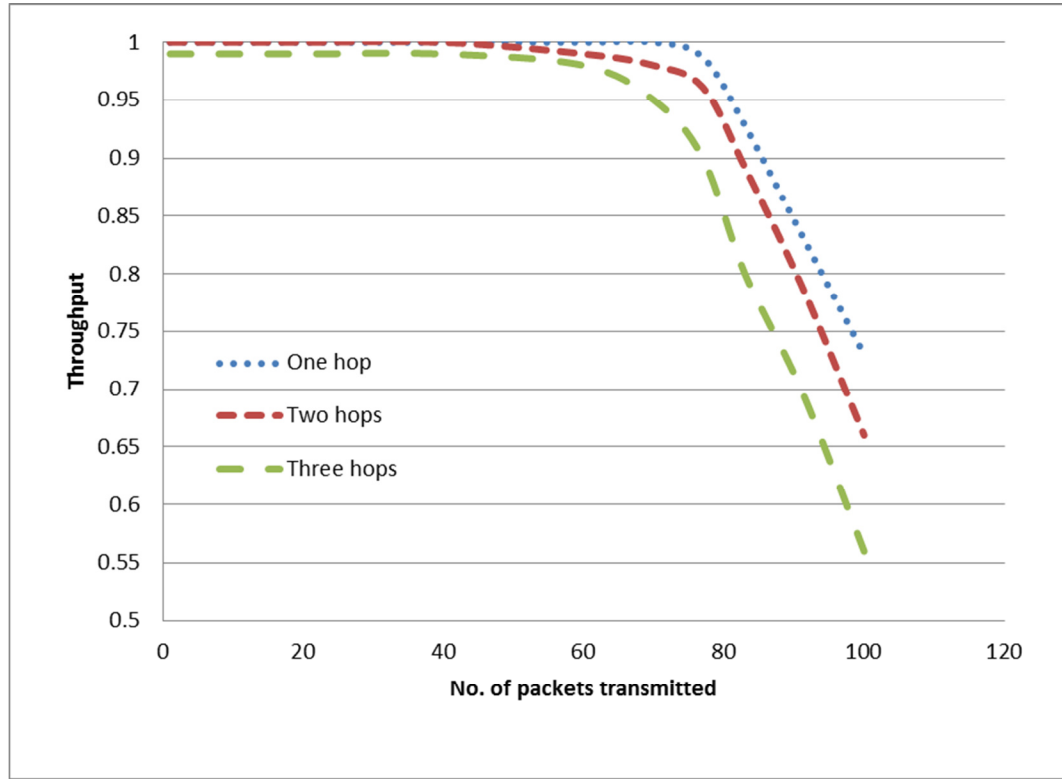


Figure 7.7: No. of packets transmitted vs. throughput.

The second experiment was conducted to study the average delay (end-to-end delay) in the network. Figure 7.8 shows that the average delay increases with the increase in number of nodes transmitting at the same time.

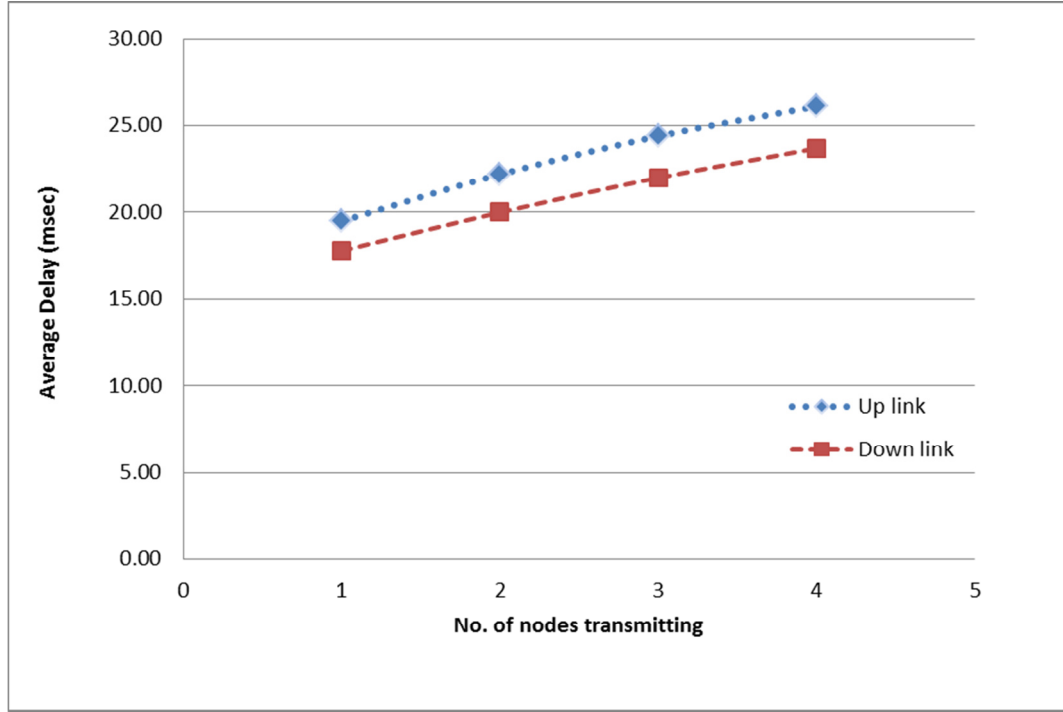


Figure 7.8: No. of nodes transmitted vs. Average delay.

7.2.5 Video Streaming over WSN

For video data to be compressed and transported over WSN, MPEG-4 (MPEG-4 Part 2) and H.264 (MPEG-4 Part 10) are the two candidate codecs. A simulation is performed to evaluate their performance in terms of peak signal-to-noise ratio (PSNR) and end-to-end delay in order to identify the best codec for video steaming in the proposed system, i.e., Sand and Dust Storm Detection System (SDSDS).

The performance of H.264 and MPEG-4 codecs is evaluated using EvalVid framework and tool-set [229–231], which is publicly available. Simulations were performed using NS2 for AODV (ad-hoc on-demand distance vector) routing protocol and CBR (constant bit rate) traffic. AODV [232] is a reactive/on-demand protocol that

starts a route discovery procedure (i.e., a global search) only when needed. It can introduce delays if routes are not instantly available in the route cache kept by nodes, and hence routes must be discovered before transmitting the data.

7.2.5.1 EvalVid Framework and Toolset

The EvalVid framework [229–231] supports video quality evaluation of the received video based on the frame-by-frame peak signal-to-noise ratio (PSNR) and quality of service (QoS) parameters of the underlying network, such as loss rates, delays and jitter. As shown in Figure 7.9, EvalVid has the following four main elements:

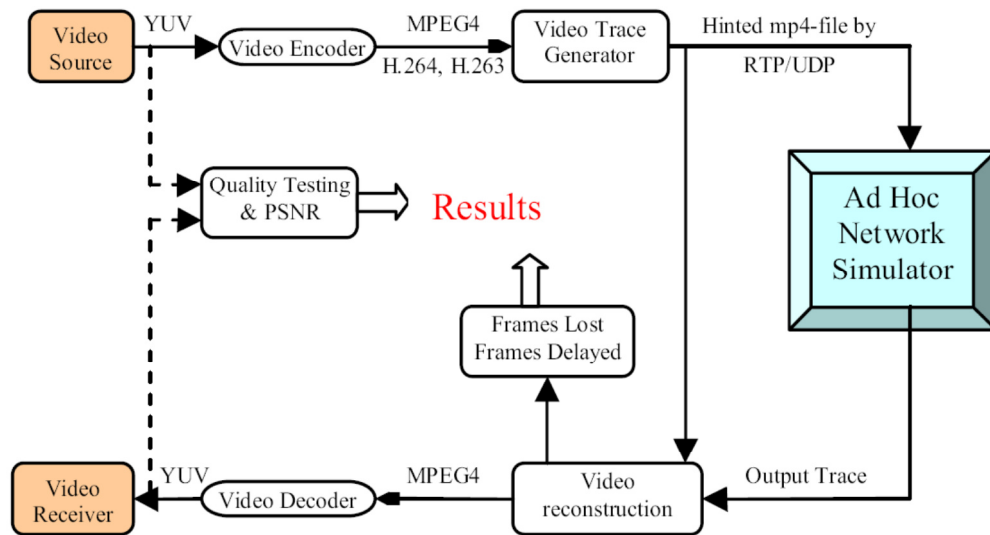


Figure 7.9: The EvalVid Framework [231].

1. Video source: It is the video file used to generate video streams for simulation. Akiyo_cif.yuv is used here, which is a YUV CIF (352 x 288) video file available online [231].
2. Video codec: It is used to convert YUV file to H.264/MPEG-4 format at the

transmitter side and convert it back to YUV format at the receiving end. H.264/MPEG-4 encoder is used here, which is available online [231].

3. Video trace generator: It reads the compressed video file which is produced by the encoder. If the frame size is larger than the upper limit, it divides each frame into smaller UDP segments before sending. A sender trace file records every outgoing UDP packet and then generates a video trace file for all frames in the actual video.
4. Video reconstruction: It reproduces the video file at the receiving end. It compares the trace files (original encoded video, video trace file, sender trace file and receiver trace file) to prepare a report on delay, loss and jitter.

7.2.5.2 Performance Parameters

To compare the performance of H.264 and MPEG-4 codecs, PSNR is used which is the most commonly used metric to check application-level QoS of video-transmission. It determines the frame-by-frame error between the reconstructed frame and the original frame using the following formulae for n^{th} frame:

$$PSNR(n)_{dB} = 20 \log_{10} \left(\frac{V_{peak}}{\sqrt{\frac{1}{N_{col} N_{row}} \sum_{i=0}^{N_{col}} \sum_{j=0}^{N_{row}} [Y_S(n, i, j) - Y_D(n, i, j)]^2}} \right) \quad (1)$$

Where S is the source frame, D is the destination frame, Y represents the luminance component, $V_{peak} = 2^k - 1$, k is the number of bits per pixel, $Y_S(n, i, j)$ and

$YD(n,i,j)$ are luminance components of n th frame, and N_{col} , N_{row} are the dimensions of the frame. The denominator is the mean square error (MSE) i.e. equation (1) can be re-written as follows:

$$PSNR(n)_{dB} = 20 \log_{10} \left(\frac{V_{peak}}{MSE} \right) \quad (2)$$

The calculation of PSNR is done by `psnr.exe` tools which generates a file containing frame-wise PSNR. Other than PSNR, the amount of data generated by both codecs and end-to-end delay for each UDP packet on the network are also calculated for analysis.

7.2.5.3 Simulation Settings

The simulations are done in Network Simulator 2 (NS2) using the parameters given in Table 7.1. The background traffic is non-multimedia UDP traffic at constant bit rate (CBR).

7.2.5.4 Experimental Results

PSNR is commonly used to assess the quality of received video. Any PSNR value greater than 37.0 means an excellent video frame while any value lesser than 20.0 is considered to be bad video frame [229], [230].

TABLE 7.1: Simulation Parameters

Parameter	Description
Simulator	ns2 version ns-allinone-2.33 [233]
Examined protocols	AODV [232]
Video file used	akiyo_cif.yuv [231]
Simulation duration	200 seconds
Channel	WirelessChannel
Mac	802.11
Antenna	OmniAntenna
Packet size	1024 Bytes
Propagation	TwoRayGround
QueueType	CMUPriQueue
Dimension	500m X 500m
Traffic type	CBR (UDP)
Packet rate	4 packets/ sec
Packet size	512 Bytes

Figure 7.10 compares the PSNR values for H.264 and MPEG-4 codecs when applied to the source video, i.e., akiyo_cif.yuv.

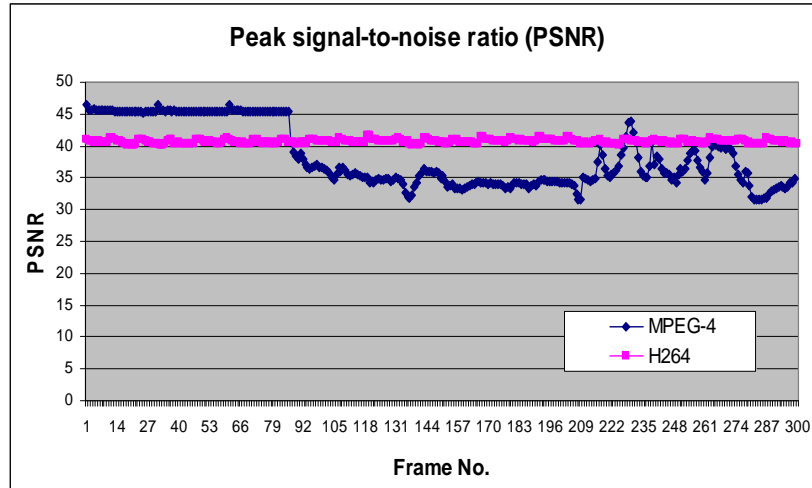


Figure 7.10: Peak signal-to-noise ratio (PSNR)

Figure 7.10 shows that the PSNR values of H.264 remain above 37 for all frames (average = 40.6992) which results in a smooth video with all excellent frames. However, PSNR of MPEG-4 remains well above 37 only for the first 93 frames and then it starts fluctuating (overall average = 38.3269) which results in distorted frames with lower PSNR. Therefore, H.264 produces a smoother video as compared to MPEG-4.

The amount of UDP data traffic on the network produced by H.264 and MPEG-4 codecs is compared in Figure 7.11. A very large amount of traffic on the network may result in congestion. A congested network may have longer queues on the intermediate nodes and higher ratio of packet dropping, and thus likely to increase the end-to-end delay.

It is evident from Figure 7.11 that the traffic generated by MPEG-4 is much higher than H.264 (i.e., about 3.5 times). Therefore, the network may face congestion and hence higher end-to-end delay for MPEG-4 when compared to H.264. Figure 7.12 shows end-to-end delay for UDP packets of the video frames generated by H.264 and MPEG-4.

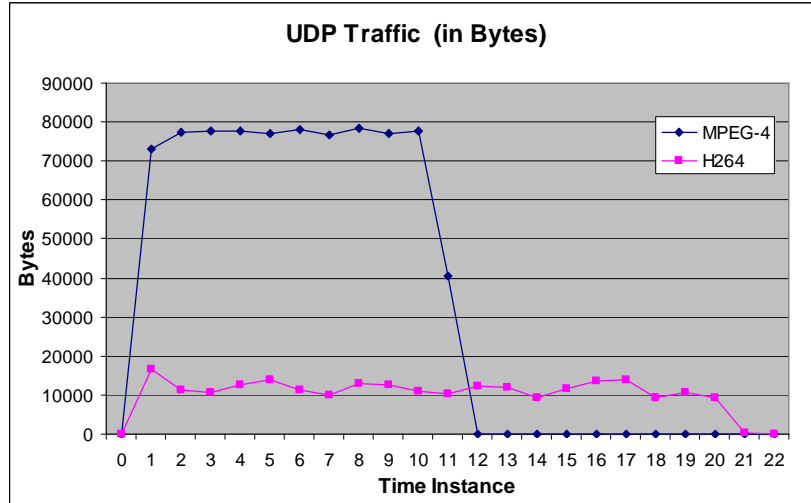


Figure 7.11: Throughput of the network

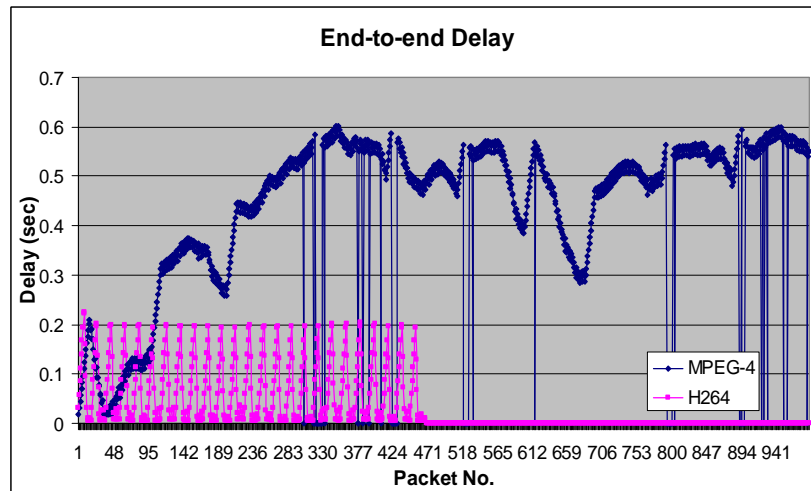


Figure 7.12: End-to-end delay

It is revealed from Figure 7.12 that MPEG-4 has higher end-to-end delay for most of the frames (overall average = 0.4428 sec). This is because of the fact that MPEG-4 generates larger frames which require more time to transmit through UDP packets on the network. In this way, H.264 is more efficient as it gives lesser end-to-end delay for most of the frames (overall average = 0.0659 sec).

These simulation results confirm that H.264 is better not only in terms of PSNR but also in terms of data traffic on the network and end-to-end delay of the data packets. These results suggest that H.264 should be used for SDSDS system because it generates lesser amount of data traffic and causes lower end-to-end delay. Moreover, H.264 videos are smoother, smaller, faster and of better quality.

7.3 Infrastructure Creation using C3 Protocol

Consider the experimental setup in Figure 7.1. The C3 algorithm is implemented and run by each node in the network. For simplicity, it is assumed that communication range (R_c) is equal to the sensing range (R_s). The sink node is attached to a notebook and is also used as the reference node for the RTSP algorithm.

Four rings are made by the C3 algorithm as shown in Figure 7.13. The current scenario shows an odd-numbered round where nodes in the odd-numbered rings have more probability to become clusterheads. However, a node in an even-numbered ring may also become clusterhead if there is no node present in its vicinity on the odd-numbered ring.

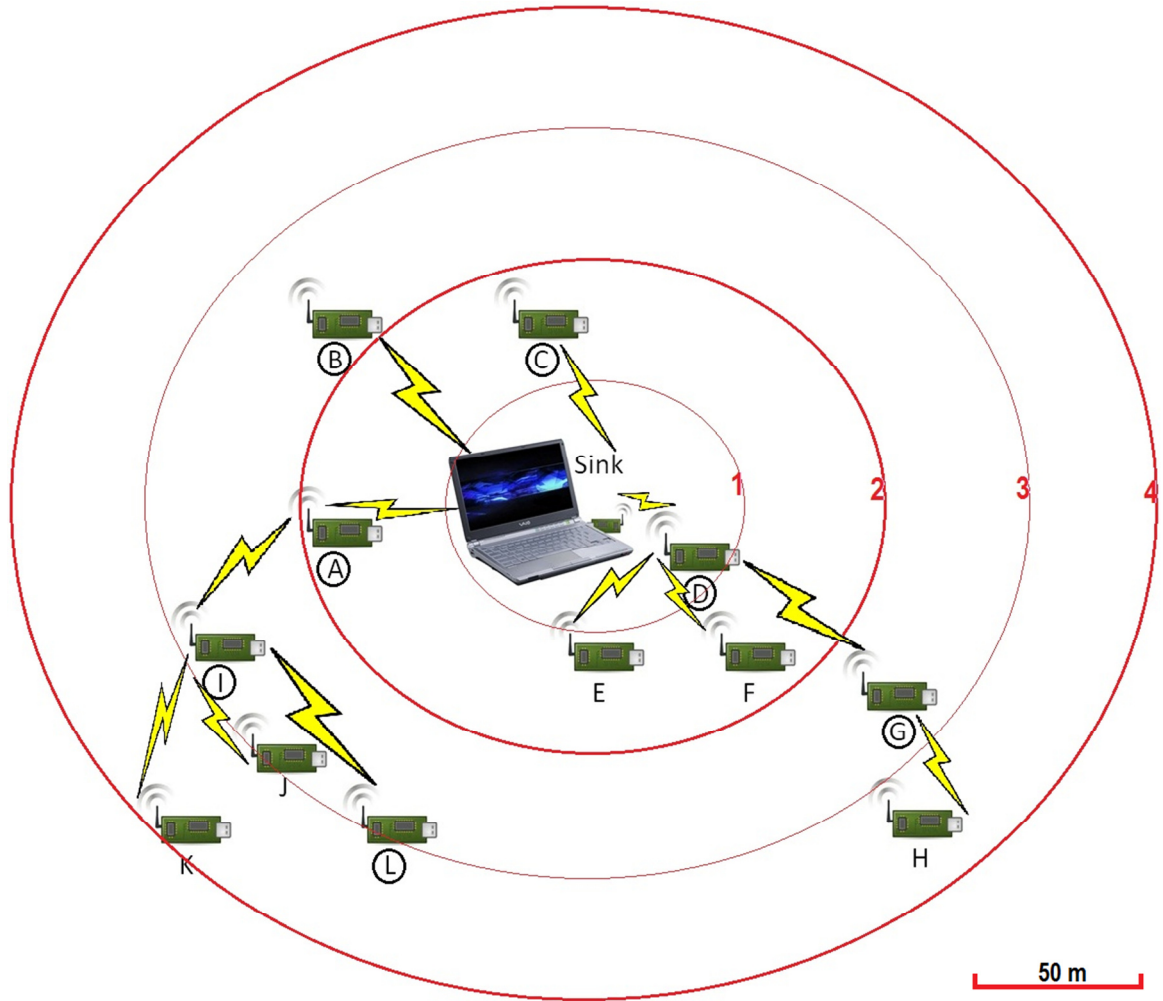


Figure 7.13: Network infrastructure created by C3 protocol.

In Figure 7.13, nodes with circled ID are elected as cluster heads. Nodes E and F are ordinary nodes which are in a cluster whose clusterhead is node D, while node H is an ordinary node which is in a cluster whose clusterhead is node G. Similarly, nodes J and K are ordinary nodes which are in a cluster whose clusterhead is node I. There are two redundant nodes in the network, i.e., node F in cluster D and node J in cluster I. These redundant nodes are put to sleep mode in the current scenario. These results validate the infrastructure creation and maintenance capabilities of C3 protocol.

7.4 Clock Synchronization using RTSP Algorithm

The proposed RTSP protocol is used for global clock synchronization in WSNs. On the other hand, FTSP is a benchmark protocol of this type. This experimental prototype implements these two algorithms only. Figure 7.14 shows the accuracy of these two algorithms on Telosb network.

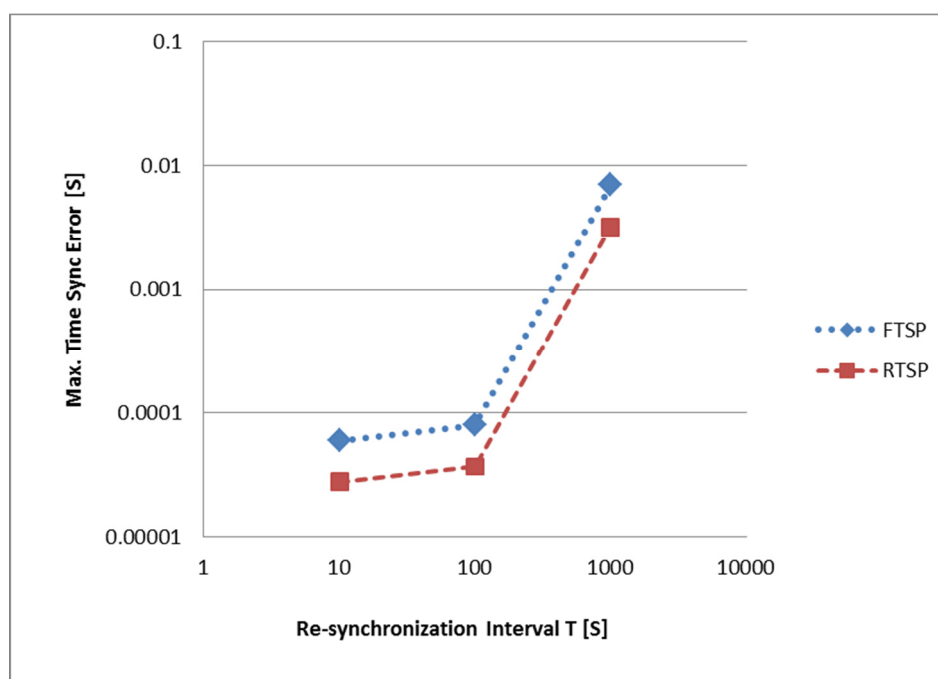


Figure 7.14: Accuracy of clock synchronization protocols.

The results of time synchronization experiment somehow match the results reported in [234]. The experimental prototype implemented only two competing algorithms for clock synchronization, i.e., FTSP and RTSP which is the proposed algorithm. The achievable accuracy for high speed timers is 0.5 μ s for FTSP and 0.23 μ s

for RTSP. As this prototype uses Telosb nodes having 32 kHz crystal timer, the achieved accuracy is much lower but comparable.

7.5 SDS Detection using Experimental Prototype

Sand and dust storm detection requires data on temperature, humidity, atmospheric pressure, wind velocity and soil moisture. The data on light, temperature and humidity is provided by sensors employed in the experimental prototype (see Figure 7.1). However, pseudo data is generated for atmospheric pressure, wind velocity, and soil moisture; and video stream for live view of the dust.

Nodes send their data to the sink attached to a laptop. Data is stored and processed there, and existence of any SDS is inferred. Due to synchronized clocks, an event can be isolated and tracked easily. The sink (or laptop) has access to the Internet and can download and process the near-real time images from Meteosat RGB Composites Dust for Eastern Africa [226]. MATLAB is used to find the ratio of red component for each pixel corresponding to the geographical area of the deployed network. A higher red ratio indicates the presence of dust. In case of any large-scale SDS in the areas closer to the WSN deployment, sink sends a message to the sensor nodes to actively read the environment; same is done in case of wind velocity above the threshold value (i.e. 6.5 m/s).

The experimental prototype successfully detected sand and dust storms of all types, i.e., small-scale SDS (covering a small geographic area), large-scale SDS

(covering a vast geographic area), short-term SDS (exists for a few minutes or hours), and long-term SDS (exists for many days or months) [92].

7.6 Conclusions

An experimental prototype is set up in order to implement and experiment the proposed algorithms. Results of several experiments validate the use of different ideas in the proposed algorithms, which include distance estimation using RSSI, synchronization using message exchange, effective coverage area by sensor nodes, throughput and average delay of the network, and video streaming over the network.

The experimental prototype also implements the proposed algorithms. The results of infrastructure creation using C3 protocol, clock synchronization using RTSP algorithm, and SDS detection using experimental prototype are very much promising.

CHAPTER 8.

CONCLUSIONS AND FUTURE DIRECTIONS

This research attempts to develop energy efficient algorithms for wireless sensor networks. It proposes a set of algorithms for clock synchronization, coverage, connectivity, communication and sandstorm detection. The main objective of these algorithms is to minimize energy-consumption in order to maximize the network lifetime.

8.1 Conclusions

First, a detailed review of the literature related to sleep-scheduling, clock synchronization, coverage, connectivity, communication and applications is provided.

Second, a comprehensive review of techniques and technologies for sand and dust storm (SDS) detection is provided. SDSs offer serious risks to the environment, economy and health. Therefore, it is very important to predict any forthcoming SDS to allow people to take any protective actions. SDS detection and prediction systems need data on dust and other environmental changes. Several technologies are available for this purpose. Satellite imaging has been commonly used for SDS detection since 1970s. However, it has some serious limitations, such as poor spatial and temporal resolution,

and inaccurate data regarding land-surface and atmosphere. In order to overcome the limitation of satellite imaging, several other alternatives have been developed such as video-surveillance, sensory information and UAV. However, these alternatives have their own limitations, such as narrow spatial coverage, unavailability of long-term data, and high cost and efforts required to establish these technologies. The solution lies in hybrid approaches which combine satellite imaging with other technologies such as video surveillance, UAV, sensory information, etc. The use of WSNs in hybrid systems is becoming popular nowadays, mainly due to lower cost and higher performance. Hybrid approaches are best suited for SDS detection and prediction because they can detect all the four types of SDS including short-term, small-scale, long-term and large-scale.

Third, a recursive time synchronization protocol (RTSP) for global time synchronization in WSNs is proposed. The RTSP gives an average accuracy of $0.23\mu\text{s}$ per hop in a large multi-hop clustered network using 7-times lesser energy than that of FTSP in the long run. An analysis of errors and efficiency is also provided. The two sources of errors in RTSP are the variation in propagation delays and relative drift between two local clocks, which are duly compensated by the algorithm. The accuracy of RTSP is enhanced by using the MAC-layer time-stamping based on SFD byte, and compensation of the propagation delay and adjustment of the timestamps at each hop. The energy-efficiency by using several techniques, which include the infrequent broadcasts by the reference node, skew-estimation using 2LR instead of 8LR, and reducing the number of time synchronization requests through the adaptive re-synchronization interval and aggregation of synchronization requests.

Fourth, a protocol for coverage, connectivity and communication (C3) in WSNs is proposed. The C3 protocol ensures partial coverage, connectivity and energy efficient communication in wireless sensor networks. It runs in four steps as follows. First, it divides the deployment area into several virtual rings of optimal width. Second, it makes clusters in the network such that the nodes at alternating rings have higher probability of being selected as clusterheads. Third, it ensures partial coverage by applying triangular tessellation inside a cluster. Finally, it allows nodes to send data through cluster heads which may involve gateways. The proposed algorithm has better performance when compared to other such protocols. However, the connectivity of WSNs depends on two things: the network density and communication range of the nodes. The partial coverage also depends on two things: the availability of nodes at required positions and the accuracy of distance estimation based on RSSI. The end-to-end delay is $1.225N$ which is slightly greater than the delay in fully active network.

Finally, a sand and dust storm detection system (SDSDS) is proposed, which is a hybrid system using WSN and satellite imaging. It uses the proposed algorithms for clock synchronization, coverage, connectivity and communication. The experimental setup consists of a WSN deployed in a small area of Arabian Desert to obtain real time data from the origin of dust, while near-real time satellite images are obtained from Meteosat website. It achieves energy efficiency by applying the proposed algorithms and by exploiting the necessary condition for SDS, i.e., a strong wind velocity, thereby transceiver and sensors on the node, except wind velocity sensor, are kept in sleeping mode until the wind velocity reaches above a threshold value. The proposed system is

highly effective and efficient due to wireless sensors providing up-to-date and accurate data from the field and satellite images requiring very simple image processing for dust detection. The proposed system achieves higher accuracy due to the middleware that ensures context fusion and conflict resolution in presence of multiple types of sensors, and makes WSN and satellite imaging support each other in detecting SDSs of all four types.

8.2 Future Directions

First, although the proposed algorithms are accurate and energy efficient, there is a room for further improvements. For example, the RTSP algorithm can be extended to provide local clock synchronization in addition to global clock synchronization. Similarly, C3 protocol can be extended to provide k-coverage and k-connectivity in the network.

Second, the proposed algorithms can be used to develop more algorithms. For example, a sleep scheduling algorithm can be developed based on the RTSP algorithm which provides high level of accuracy. Similarly, a coarse-grain localization scheme can be developed based on the C3 protocol which divides the network into several virtual regions.

Finally, the SDSDS system can be extended to detect other environmental changes in addition to sand and dust storms. A multisensory fusion approach is planned for full-fledge implementation of the system in near future.

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [3] S. DiStasi, C. P. Townsend, J. Galbreath, and S. W. Arms, “Scalable, synchronized, energy harvesting wireless sensor networks,” in *Prognostics and Health Management Conference, 2010. PHM’10.*, 2010, pp. 1–5.
- [4] S. W. Arms, C. P. Townsend, D. L. Churchill, J. H. Galbreath, and S. W. Mundell, “Power management for energy harvesting wireless sensors,” in *Proc. SPIE*, 2005, vol. 5763, pp. 267–275.
- [5] L. Mateu, F. B. Moll Echeto, and others, “Review of energy harvesting techniques and applications for microelectronics,” 2011.
- [6] E. Velazquez and N. Santoro, “Efficient robot-based energy maintenance in Wireless Sensor Networks,” in *Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on*, 2010, pp. 313–318.
- [7] G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton, “Sensor network-based countersniper system,” in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 1–12.

- [8] J. Yick, B. Mukherjee, and D. Ghosal, "Analysis of a prediction-based mobility adaptive tracking algorithm," in *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*, 2005, pp. 753–760.
- [9] M. Castillo-Effer, D. H. Quintela, W. Moreno, R. Jordan, and W. Westhoff, "Wireless sensor networks for flash-flood alerting," in *Devices, Circuits and Systems, 2004. Proceedings of the Fifth IEEE International Caracas Conference on*, 2004, vol. 1, pp. 142–146.
- [10] T. Gao, D. Greenspan, M. Welsh, R. Juang, and A. Alm, "Vital signs monitoring and patient tracking over a wireless network," in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, 2006, pp. 102–105.
- [11] K. Lorincz, D. J. Malan, T. R. . Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton, "Sensor networks for emergency response: Challenges and opportunities," *Pervasive Computing, IEEE*, vol. 3, no. 4, pp. 16–23, 2004.
- [12] G. Wener-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Walsh, "Deploying a wireless sensor network on an active volcano. Data-Driven Applications in Sensor Networks (Special Issue)," *IEEE Internet Computing*, vol. 2, pp. 18–25, 2006.
- [13] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson, "Analysis of wireless sensor networks for habitat monitoring," *Wireless sensor networks*, pp. 399–423, 2004.

- [14] W. Manges, G. Allgood, and S. Smith, "It's time for sensors to go wireless. part 2: Take a good technology and make it an economic success," *Sensors Magazine*, vol. 5, 1999.
- [15] X. J. Hao and J. J. Qu, "Saharan dust storm detection using moderate resolution imaging spectroradiometer thermal infrared bands," *Journal of Applied Remote Sensing*, vol. 1, no. 013510, p. 013510, 2007.
- [16] B. Krishnamachari, *Networking Wireless Sensors*. New York: Cambridge University Press, 2005.
- [17] S. Singh and C. S. Raghavendra, "PAMAS—power aware multi-access protocol with signalling for ad hoc networks," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 3, pp. 5–26, 1998.
- [18] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on wireless: An event driven energy saving strategy for battery operated devices," in *Proceedings of the 8th annual international conference on Mobile computing and networking*, 2002, pp. 160–171.
- [19] J. L. Hill and D. E. Culler, "Mica: A wireless platform for deeply embedded networks," *Micro, IEEE*, vol. 22, no. 6, pp. 12–24, 2002.
- [20] A. El-Hoiydi, "Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks," in *Communications, 2002. ICC 2002. IEEE International Conference on*, 2002, vol. 5, pp. 3418–3423.
- [21] A. El-Hoiydi and J. D. Decotignie, "WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks," *Algorithmic Aspects of Wireless Sensor Networks*, pp. 18–31, 2004.

- [22] E. Y. . Lin, J. M. Rabaey, and A. Wolisz, "Power-efficient rendez-vous schemes for dense wireless sensor networks," in *Communications, 2004 IEEE International Conference on*, 2004, vol. 7, pp. 3769–3776.
- [23] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 95–107.
- [24] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2002, vol. 3, pp. 1567–1576.
- [25] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *Networking, IEEE/ACM Transactions on*, vol. 12, no. 3, pp. 493–506, 2004.
- [26] T. Van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 171–180.
- [27] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," in *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, 2004, p. 224.
- [28] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, "Delay efficient sleep scheduling in wireless sensor networks," in *INFOCOM 2005. 24th Annual Joint*

Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, 2005, vol. 4, pp. 2470–2481.

- [29] R. Zheng, J. C. Hou, and L. Sha, “Asynchronous wakeup for ad hoc networks,” in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, 2003, pp. 35–45.
- [30] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, “Protocols for self-organization of a wireless sensor network,” *Personal Communications, IEEE*, vol. 7, no. 5, pp. 16–27, 2000.
- [31] W. S. Conner, J. Chhabra, M. Yarvis, and L. Krishnamurthy, “Experimental evaluation of synchronization and topology control for in-building sensor network applications,” in *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, 2003, pp. 38–49.
- [32] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, “Energy-efficient, collision-free medium access control for wireless sensor networks,” *Wireless Networks*, vol. 12, no. 1, pp. 63–78, 2006.
- [33] V. Rodoplu and T. H. Meng, “Minimum energy mobile wireless networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 8, pp. 1333–1344, 1999.
- [34] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, “Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol,” in *European Wireless Conference*, 2002, vol. 2002.
- [35] R. Ramanathan and R. Rosales-Hain, “Topology control of multihop wireless networks using transmit power adjustment,” in *INFOCOM 2000. Nineteenth*

- Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2000, vol. 2, pp. 404–413.
- [36] R. Wattenhofer, L. Li, P. Bahl, and Y. M. Wang, “Distributed topology control for power efficient operation in multihop wireless ad hoc networks,” in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2001, vol. 3, pp. 1388–1397.
- [37] L. Li, J. Y. Halpern, P. Bahl, Y. M. Wang, and R. Wattenhofer, “Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks,” in *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, 2001, pp. 264–273.
- [38] N. Li, J. C. Hou, and L. Sha, “Design and analysis of an MST-based topology control algorithm,” in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, 2003, vol. 3, pp. 1702–1712.
- [39] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, “Integrated coverage and connectivity configuration in wireless sensor networks,” in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 28–39.
- [40] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, “Coverage problems in wireless ad-hoc sensor networks,” in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2001, vol. 3, pp. 1380–1387.

- [41] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, 2002, pp. 299–308.
- [42] N. Heo and P. K. Varshney, "An intelligent deployment and clustering algorithm for a distributed mobile sensor network," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, 2003, vol. 5, pp. 4576–4581.
- [43] A. Howard, M. J. Mataric, and G. S. Sukhatme, "An incremental deployment algorithm for mobile robot teams," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, 2002, vol. 3, pp. 2849–2854.
- [44] G. Wang, G. Cao, and T. LaPorta, "A bidding protocol for deploying mobile sensors," in *Network Protocols, 2003. Proceedings. 11th IEEE International Conference on*, 2003, pp. 315–324.
- [45] M. A. Batalin and G. S. Sukhatme, "Coverage, exploration and deployment by a mobile robot and communication network," *Telecommunication Systems*, vol. 26, no. 2, pp. 181–196, 2004.
- [46] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme, "Deployment and connectivity repair of a sensor net with a flying robot," *Experimental Robotics IX*, pp. 333–343, 2006.
- [47] Y. Xu, J. Heidemann, and D. Estrin, "Adaptive energy-conserving routing for multihop ad hoc networks," in *Research Report 527, USC/Information Sciences Institute*, 2000.

- [48] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001, pp. 70–84.
- [49] A. Cerpa and D. Estrin, "ASCENT: Adaptive self-configuring sensor networks topologies," *mobile computing, IEEE transactions on*, vol. 3, no. 3, pp. 272–285, 2004.
- [50] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 481–494, 2002.
- [51] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "PEAS: A robust energy conserving protocol for long-lived sensor networks," in *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on*, 2003, pp. 28–37.
- [52] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, 2002, pp. 32–41.
- [53] K. Wu, Y. Gao, F. Li, and Y. Xiao, "Lightweight deployment-aware scheduling for wireless sensor networks," *Mobile networks and applications*, vol. 10, no. 6, pp. 837–852, 2005.
- [54] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 14–27.

- [55] A. E. Khandani, J. Abounadi, E. Modiano, and L. Zheng, “Reliability and route diversity in wireless networks,” *Wireless Communications, IEEE Transactions on*, vol. 7, no. 12, pp. 4772–4776, 2008.
- [56] S. Biswas and R. Morris, “Opportunistic routing in multi-hop wireless networks,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 1, pp. 69–74, 2004.
- [57] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, “Highly-resilient, energy-efficient multipath routing in wireless sensor networks,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 11–25, 2001.
- [58] R. Poor, C. Bowman, and C. B. Auburn, “Self-healing networks,” *Queue*, vol. 1, no. 3, pp. 52–59, 2003.
- [59] F. Ye, G. Zhong, S. Lu, and L. Zhang, “A robust data delivery protocol for large scale sensor networks,” in *Proceedings of the 2nd international conference on Information processing in sensor networks*, 2003, pp. 658–673.
- [60] S. Singh, M. Woo, and C. S. Raghavendra, “Power-aware routing in mobile ad hoc networks,” in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, 1998, pp. 181–190.
- [61] J. H. Chang and L. Tassiulas, “Energy conserving routing in wireless ad-hoc networks,” in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2000, vol. 1, pp. 22–31.

- [62] R. C. Shah and J. M. Rabaey, “Energy aware routing for low energy ad hoc sensor networks,” in *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, 2002, vol. 1, pp. 350–355.
- [63] J. H. Chang and L. Tassiulas, “Maximum lifetime routing in wireless sensor networks,” *Networking, IEEE/ACM Transactions on*, vol. 12, no. 4, pp. 609–619, 2004.
- [64] G. G. Finn, “Routing and addressing problems in large metropolitan-scale internetworks,” DTIC Document, 1987.
- [65] H. Takagi and L. Kleinrock, “Optimal transmission ranges for randomly distributed packet radio terminals,” *Communications, IEEE Transactions on*, vol. 32, no. 3, pp. 246–257, 1984.
- [66] T. C. Hou and V. Li, “Transmission range control in multihop packet radio networks,” *Communications, IEEE Transactions on*, vol. 34, no. 1, pp. 38–44, 1986.
- [67] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia, “Routing with guaranteed delivery in ad hoc wireless networks,” *Wireless Networks*, vol. 7, no. 6, pp. 609–616, 2001.
- [68] B. Karp and H. T. Kung, “GPSR: Greedy perimeter stateless routing for wireless networks,” in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 243–254.
- [69] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari, “Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks,” in

Proceedings of the 2nd international conference on Embedded networked sensor systems, 2004, pp. 108–121.

- [70] Y. Yu, R. Govindan, and D. Estrin, “Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks,” Technical report ucla/csd-tr-01-0023, UCLA Computer Science Department, 2001.
- [71] D. Niculescu and B. Nath, “Trajectory based forwarding and its applications,” in *Proceedings of the 9th annual international conference on Mobile computing and networking*, 2003, pp. 260–272.
- [72] T. Sheltami and H. Mouftah, “Virtual base station on-demand,” in *The 2002 International Conference on Wireless Networks (ICWN’02) June*, 2002, vol. 24, p. 27.
- [73] T. Sheltami and H. T. Mouftah, “An efficient energy aware clusterhead formation infrastructure protocol for MANETs,” in *Computers and Communication, 2003.(ISCC 2003). Proceedings. Eighth IEEE International Symposium on*, 2003, pp. 203–208.
- [74] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388–404, 2000.
- [75] S. Doshi and T. X. Brown, “Minimum energy routing schemes for a wireless ad hoc network,” in *IEEE INFOCOM*, 2002, vol. 2.
- [76] M. Ettus, “System capacity, latency, and power consumption in multihop-routed SS-CDMA wireless networks,” in *Radio and Wireless Conference, 1998. RAWCON 98. 1998 IEEE*, 1998, pp. 55–58.

- [77] J. Kuruvila, A. Nayak, and I. Stojmenovic, "Progress based localized power and cost aware routing algorithms for ad hoc and sensor wireless networks," *Ad-Hoc, Mobile, and Wireless Networks*, pp. 630–630, 2004.
- [78] S. Narayanaswamy, V. Kawadia, R. Sreenivas, and P. Kumar, "The COMPOW protocol for power control in ad hoc networks: Theory, architecture, algorithm, implementation, and experimentation," *Urbana*, vol. 51, p. 61801, 2001.
- [79] T. Sheltami and H. Mouftah, "Performance Comparison of COMPOW and Minimum Energy Routing Protocols for Sensor Networks," in *the 22nd Biennial Symposium on Communications*, Canada, 2004, pp. 232–235.
- [80] V. Kawadia and P. Kumar, "Power control and clustering in ad hoc networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, 2003, vol. 1, pp. 459–469.
- [81] T. R. Sheltami, "Neighbor-Aware Clusterhead for SNET," in *Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on*, 2008, pp. 561–566.
- [82] H. S. Kim, T. F. Abdelzaher, and W. H. Kwon, "Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 193–204.
- [83] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Networks*, vol. 1, no. 2–3, pp. 215–233, 2003.

- [84] P. Baruah, R. Urgaonkar, and B. Krishnamachari, "Learning-enforced time domain routing to mobile sinks in wireless sensor fields," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, 2004, pp. 525–532.
- [85] R. Tsolmon, L. Ochirkhuyag, and T. Sternberg, "Monitoring the source of transnational dust storms in north east Asia," *International Journal of Digital Earth*, vol. 1, no. 1, pp. 119–129, 2008.
- [86] K. Gyan, W. Henry, S. Lacaille, A. Laloo, C. Lamsee-Ebanks, S. McKay, R. Antoine, and M. Monteil, "African dust clouds are associated with increased paediatric asthma accident and emergency admissions on the Caribbean island of Trinidad," *International journal of biometeorology*, vol. 49, no. 6, pp. 371–376, 2005.
- [87] B. Ogorodnikov, "A dust storm over the Ukraine and Belarus territory contaminated by radionuclides after the Chernobyl accident," *Russian Meteorology and Hydrology*, vol. 36, no. 9, pp. 613–623, 2011.
- [88] J. Slanina, "Air pollution: the emission–effect relation," *Reviews in Environmental Science and Biotechnology*, vol. 6, no. 4, pp. 353–374, 2007.
- [89] W. Zhang, X. Liu, W. Xiao, and D. Chi, "Software design of sand-dust storm warning system based on grey correlation analysis and particle swarm optimization support vector machine," in *Proceedings of the 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS)*, 2009, vol. 2, pp. 47–50.

- [90] J. Y. Lim and Y. Chun, “The characteristics of Asian dust events in Northeast Asia during the springtime from 1993 to 2004,” *Global and Planetary Change*, vol. 52, no. 1–4, pp. 231–247, 2006.
- [91] Y. Shao, *Physics and Modelling of Wind Erosion*, 2nd ed., vol. 37. Netherlands: Springer, 2008.
- [92] T. Gao and J. Han, “Evolutionary characteristics of the atmospheric circulations for frequent and infrequent dust storm springs in northern China and the detection of potential future seasonal forecast signals,” *Meteorological Applications*, vol. 17, no. 1, pp. 76–87, 2010.
- [93] WMO, “Climate and Land Degradation,” World Meteorological Organization, Switzerland, WMO-No. 989, 2005.
- [94] H. El-Askary, M. Kafatos, X. Liu, and T. El-Ghazawi, “Introducing new approaches for dust storms detection using remote sensing technology,” in *Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2003, vol. 4, pp. 2439–2441.
- [95] Y. Ma, W. Gong, P. Wang, and X. Hu, “New dust aerosol identification method for spaceborne lidar measurements,” *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 112, no. 2, pp. 338 – 345, 2011.
- [96] J. Comiso, “Chapter 3: Satellite Remote Sensing Techniques,” in *Polar Oceans from Space*, vol. 41, New York: Springer, 2010, pp. 73–111.
- [97] J. Huang, J. Ge, and F. Weng, “Detection of Asia dust storms using multisensor satellite measurements,” *Remote Sensing of Environment*, vol. 110, no. 2, pp. 186–191, 2007.

- [98] L. San-chao, L. Qinhua, G. Maofang, and C. Liangfu, "Detection of Dust Storms by Using Daytime and Nighttime Multi-spectral MODIS Images," in *Proceedings of the IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS)*, 2006, pp. 294–296.
- [99] H. El-Askary, R. Gautam, R. P. Singh, and M. Kafatos, "Dust storms detection over the Indo-Gangetic basin using multi sensor data," *Advances in Space Research*, vol. 37, no. 4, pp. 728–733, 2006.
- [100] W. Fu-cun, F. U. You-zhi, and L. I. Hong, "Diagnostic Analysis and Weather Radar Observation of a Sand-dust Storm in Autumn," *Journal of Desert Research*, vol. 28, no. 1, pp. 170–177, 2008.
- [101] D. G. Kaskaoutis, H. D. Kambezidis, P. T. Nastos, and P. G. Kosmopoulos, "Study on an intense dust storm over Greece," *Atmospheric Environment*, vol. 42, no. 29, pp. 6884–6896, 2008.
- [102] P. Wang, Z. Sun, M. C. Vuran, M. A. Al-Rodhaan, A. M. Al-Dhelaan, and I. F. Akyildiz, "On network connectivity of wireless sensor networks for sandstorm monitoring," *Computer Networks*, vol. 55, no. 5, pp. 1150–1157, 2011.
- [103] Q. Zeng, X. Cheng, F. Hu, and Z. Peng, "Gustiness and coherent structure of strong winds and their role in dust emission and entrainment," *Advances in Atmospheric Sciences*, vol. 27, no. 1, pp. 1–13, 2010.
- [104] D. A. Gillette and K. J. Hanson, "Spatial and temporal variability of dust production caused by wind erosion in the United States," *Journal of Geophysical Research*, vol. 94, no. D2, pp. 2197–2206, 1989.

- [105] D. A. Gillette and R. Passi, “Modeling dust emission caused by wind erosion,” *Journal of Geophysical Research*, vol. 93, no. D11, pp. 14233–14, 1988.
- [106] S. Joussaume, “Three-dimensional simulations of the atmospheric cycle of desert dust particles using a general circulation model,” *Journal of Geophysical Research*, vol. 95, no. D2, pp. 1909–1941, 1990.
- [107] D. L. Westphal, O. B. Toon, and T. N. Carson, “A case study of mobilisation and transport of Saharan dust,” *Journal of the Atmospheric Sciences*, vol. 45, pp. 2145–2175, 1988.
- [108] P. Ginoux, “Sources and distributions of dust aerosols simulated with the GOCART model,” *Journal of Geophysical Research D: Atmospheres*, vol. 106, no. D17, pp. 20255–20273, 2001.
- [109] P. Ginoux, J. M. Prospero, O. Torres, and M. Chin, “Long-term simulation of global dust distribution with the GOCART model: correlation with North Atlantic Oscillation,” *Environmental Modelling & Software*, vol. 19, no. 2, pp. 113–128, 2004.
- [110] T. Y. Tanaka, K. Orito, T. T. Sekiyama, K. Shibata, M. Chiba, and H. Tanaka, “MASINGAR, a global tropospheric aerosol chemical transport model coupled with MRI/JMA 98 GCM- Model description,” *Papers in Meteorology and Geophysics*, vol. 53, no. 4, pp. 119–138, 2003.
- [111] I. Tegen, “Modeling of mineral dust in the atmosphere: sources, transport, and optical thickness,” *Journal of Geophysical Research*, vol. 99, no. D11, pp. 22,897–22,914, 1994.

- [112] C. S. Zender, H. Bian, and D. Newman, “Mineral Dust Entrainment and Deposition (DEAD) model: Description and 1990s dust climatology,” *Journal of Geophysical Research*, vol. 108, no. D14, p. 4416, 2003.
- [113] Y. Ashkenazy, H. Yizhaq, and H. Tsoar, “Sand dune mobility under climate change in the Kalahari and Australian deserts,” *Climatic Change*, pp. 1–23, 2011.
- [114] S. L. Gong, “Characterization of soil dust aerosol in China and its transport and distribution during 2001 ACE-Asia: 2. Model simulation and validation,” *Journal of Geophysical Research D: Atmospheres*, vol. 108, no. 9, pp. ACH 4–1 – ACH 4–19, 2003.
- [115] R. B. Husar, D. M. Tratt, B. Schichtel, S. R. Falke, F. Li, D. Jaffe, S. Gasso, T. Gill, N. S. Laulainen, F. Lu, and others, “Asian dust events of April 1998,” *Journal of Geophysical Research*, vol. 106, no. D16, pp. 18317–18330, 2001.
- [116] M. Liu, “A high-resolution numerical study of the Asian dust storms of April 2001,” *Journal of Geophysical Research D: Atmospheres*, vol. 108, no. 23, pp. ACE 21–1 – ACE 21–21, 2003.
- [117] S. U. Park, A. Choe, and M. S. Park, “A simulation of Asian dust events in March 2010 by using the ADAM2 model,” *Theoretical and Applied Climatology*, pp. 1–13, 2011.
- [118] Y. Shao and L. M. Leslie, “Wind erosion prediction over the Australian continent,” *Journal of Geophysical Research*, vol. 102, no. D25, pp. 30091–30105, 1997.

- [119] I. Uno, H. Amano, S. Emori, K. Kinoshita, I. Matsui, and N. Sugimoto, “Trans-Pacific yellow sand transport observed in April 1998: A numerical simulation,” *Journal of Geophysical Research*, vol. 106, no. D16, pp. 18331–18, 2001.
- [120] Z. Wang, H. Ueda, and M. Huang, “A deflation module for use in modeling long-range transport of yellow sand over East Asia,” *Journal of Geophysical Research*, vol. 105, no. D22, pp. 26947–26959, 2000.
- [121] N. Seino, H. Sasaki, A. Yamamoto, M. Mikami, H. Zhou, and F. Zeng, “Numerical simulation of mesoscale circulations in the tarim basin associated with dust events,” *Journal of the Meteorological Society of Japan*, vol. 83, pp. 205–218, 2005.
- [122] I. Uno, K. Harada, S. Satake, Y. Hara, and Z. Wang, “Meteorological characteristics and dust distribution of the Tarim Basin simulated by the nesting RAMS/CFORS dust model,” *Journal of the Meteorological Society of Japan*, vol. 83, pp. 219–239, 2005.
- [123] Y. Shao and C. H. Dong, “A review on East Asian dust storm climate, modelling and monitoring,” *Global and Planetary Change*, vol. 52, no. 1–4, pp. 1–22, 2006.
- [124] Y. Shao, Y. Yang, J. J. Wang, Z. X. Song, L. M. Leslie, C. H. Dong, S. H. Zhang, Z. H. Lin, Y. Kanai, S. Yabuki, and others, “Real-time numerical prediction of northeast Asian dust storms using an integrated modeling system,” *Journal of Geophysical Research*, vol. 108, p. 4691, 2003.
- [125] J. Sun, L. Zhao, S. Zhao, and R. Zhang, “An integrated dust storm prediction system suitable for east Asia and its simulation results,” *Global and Planetary Change*, vol. 52, no. 1–4, pp. 71–87, 2006.

- [126] B. Marticorena, G. Bergametti, D. Gillette, and J. Belnap, "Factors controlling threshold friction velocity in semiarid and arid areas of the United States," *Journal of Geophysical Research*, vol. 102, no. 23, pp. 277–23, 1997.
- [127] P. S. Chavez Jr, D. J. Mackinnon, R. L. Reynolds, and M. Velasco, "Monitoring dust storms and mapping landscape vulnerability to wind erosion using satellite and ground-based digital images," *Geomorphology*, vol. 51, pp. 1–8, 2002.
- [128] A. G. Wain, S. Lee, G. A. Mills, G. D. Hess, M. E. Cope, and N. Tindale, "Meteorological overview and verification of HYSPLIT and AAQFS dust forecasts for the duststorm of 22-24 October 2002," *Australian Meteorological Magazine*, vol. 55, no. 1, pp. 35–46, 2006.
- [129] Y. Shao, "Simplification of a dust emission scheme and comparison with data," *Journal of Geophysical Research*, vol. 109, no. D10, p. D10202, 2004.
- [130] Y. Shao, "A model for mineral dust emission," *Journal of Geophysical Research*, vol. 106, no. D17, pp. 20239–20254, 2001.
- [131] A. E. Wald, Y. J. Kaufman, D. Tanré, and B. C. Gao, "Daytime and nighttime detection of mineral dust over desert using infrared spectral contrast," *Journal of Geophysical Research*, vol. 103, no. D24, pp. 32307–32, 1998.
- [132] Z. Lu, J. Dai, Y. Yang, and H. Liu, "Research on the Forecasting Model of Sand-Dust Storm Based on the Grid Field," in *Proceedings of the Third International Conference on Natural Computation*, 2007, vol. 3, pp. 348–352.
- [133] E. A. . Elsheikh, M. Islam, A. H. M. Alam, A. F. Ismail, K. Al-Khateeb, and Z. Elabdin, "The effect of particle size distributions on dust storm attenuation prediction for microwave propagation," in *Proceedings of the International*

- Conference on Computer and Communication Engineering (ICCCE)*, Kuala Lumpur, 2010, pp. 1–5.
- [134] L. Natsagdorj, D. Jugder, and Y. S. Chung, “Analysis of dust storms observed in Mongolia during 1937-1999,” *Atmospheric Environment*, vol. 37, no. 9–10, pp. 1401–1411, 2003.
- [135] M. Legrand, A. Plana-Fattori, and C. N’doum, “Satellite detection of dust using the IR imagery of Meteosat,” *Journal of Geophysical Research*, vol. 106, no. D16, pp. 18–251, 2001.
- [136] J. J. Qu, X. Hao, M. Kafatos, and L. Wang, “Asian dust storm monitoring combining Terra and Aqua MODIS SRB measurements,” *Geoscience and Remote Sensing Letters, IEEE*, vol. 3, no. 4, pp. 484–486, 2006.
- [137] S. L. Gong and X. Y. Zhang, “CUACE/Dust - an integrated system of observation and modeling systems for operational dust forecasting in Asia,” *Atmospheric Chemistry and Physics Discussions*, vol. 7, no. 4, pp. 10323–10342, 2007.
- [138] S. U. Park and H. J. In, “Parameterization of dust emission for the simulation of the Yellow Sand (Asian dust) event observed in March 2002 in Korea,” *Journal of Geophysical Research*, vol. 108, no. D19, p. 4618, 2003.
- [139] S. Yaping, Y. Yan, W. Jianjie, and others, “Northeast Asian dust storms: Real-time numerical prediction and validation,” *Journal of Geophysical Research*, vol. 108, pp. 4691–47081, 2003.
- [140] C. H. Zhou, S. L. Gong, X. Y. Zhang, Y. Q. Wang, T. Niu, H. L. Liu, T. L. Zhao, Y. Q. Yang, and Q. Hou, “Development and evaluation of an operational SDS

- forecasting system for East Asia: CUACE/Dust,” *Atmospheric Chemistry and Physics*, vol. 8, no. 4, pp. 787–798, 2008.
- [141] X. Q. Hu, N. M. Lu, T. Niu, and P. Zhang, “Operational retrieval of Asian sand and dust storm from FY-2C geostationary meteorological satellite and its application to real time forecast in Asia,” *Atmospheric Chemistry and Physics*, vol. 8, no. 6, pp. 1649–1659, 2008.
- [142] W. Wang and Z. Y. Fang, “Numerical simulation and synoptic analysis of dust emission and transport in East Asia,” *Global and Planetary Change*, vol. 52, no. 1–4, pp. 57–70, 2006.
- [143] J. Dulam, “Discriminate analysis for dust storm prediction in the Gobi and steppe regions in Mongolia,” *Water, Air, & Soil Pollution: Focus*, vol. 5, no. 3, pp. 37–49, 2005.
- [144] G. Tao, L. Jingtao, Y. Xiao, K. Ling, F. Yida, and H. Yinghua, “Objective pattern discrimination model for dust storm forecasting,” *Meteorological Applications*, vol. 9, no. 01, pp. 55–62, 2002.
- [145] Y. Q. Yang, J. Z. Wang, Q. Hou, Y. Li, and C. H. Zhou, “Discriminant Genetic Algorithm Extended (DGAE) model for seasonal sand and dust storm prediction,” *SCIENCE CHINA Earth Sciences*, pp. 1–9, 2011.
- [146] M. Chacon-Murguía, Y. Quezada-Holguín, P. Rivas-Perea, and S. Cabrera, “Dust storm detection using a neural network with uncertainty and ambiguity output analysis,” *Pattern Recognition*, pp. 305–313, 2011.

- [147] M. Huang, G. Peng, J. Zhang, and S. Zhang, "Application of artificial neural networks to the prediction of dust storms in Northwest China," *Global and Planetary change*, vol. 52, no. 1–4, pp. 216–224, 2006.
- [148] X. Lang, "Seasonal prediction of spring dust weather frequency in Beijing," *Acta Meteorologica Sinica*, vol. 25, no. 5, pp. 682–690, 2011.
- [149] J. R. Jiang, X. Y. Zhang, C. H. Zhou, H. L. Liu, and X. B. Chi, "Study on Sand-dust Model Coupled with PSU/NCAR Mesoscale Model and Its Application in Northeast Asia," in *Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, 2010, pp. 129–134.
- [150] Z. Qingcun, L. Zhaohui, and Z. Guangqing, "Dynamical extra-seasonal climate prediction system IAP DCP-II," *Chinese Journal of Atmospheric Sciences*, vol. 27, no. 2, pp. 101–117, 2003.
- [151] Q. C. Zeng, C. G. Yuan, W. Q. Wang, and R. H. Zhang, "Experiments in numerical extra-seasonal prediction of climate anomalies," *Chinese Journal of Atmospheric Sciences*, vol. 14, pp. 10–25, 1990.
- [152] B. H. Barnum, N. S. Winstead, J. Wesely, A. Hakola, P. R. Colarco, O. B. Toon, P. Ginoux, G. Brooks, L. Hasselbarth, and B. Toth, "Forecasting dust storms using the CARMA-dust model and MM5 weather data," *Environmental Modelling & Software*, vol. 19, no. 2, pp. 129–140, 2004.
- [153] S. Nickovic, G. Kallos, A. Papadopoulos, and O. Kakaliagou, "A model for prediction of desert dust cycle in the atmosphere," *Journal of Geophysical Research*, vol. 106, no. D16, pp. 18113–18129, 2001.

- [154] Y. B. Wang, J. F. Fei, X. G. Huang, X. P. Cheng, and Y. J. Ge, “Application of Models-3/CMAQ on Asian Dust Storm,” *Advanced Materials Research*, vol. 378, pp. 385–388, 2012.
- [155] K. Hansen, “NASA-Enhanced Dust Storm Predictions To Aid Health Community,” 28-Oct-2008. [Online]. Available: <http://www.nasa.gov/topics/earth/features/phairs.html>. [Accessed: 31-Jan-2012].
- [156] P. Ginoux, M. Chin, I. Tegen, J. Prospero, B. Holben, O. Dubovik, and S. J. Lin, “Global simulation of dust in the troposphere: Model description and assessment,” *J. Geophys. Res*, vol. 106, no. 20, pp. 255–20, 2001.
- [157] S. Engelstaedter, “Dust storm frequencies and their relationship to land surface conditions,” Diploma thesis, Inst. Geosci. Friedrich-Schiller-University, Jena, Germany, 2001.
- [158] D. Martell, “Forest fire management,” in *Handbook of Operations Research in Natural Resources*, vol. 99, New York: Springer, 2007, pp. 489–509.
- [159] B. S. Abdulelah, “Depicting Dust and Sand Storm Signatures Through the Means of Satellite Images and Ground-Based Observations for Saudi Arabia,” PhD Thesis, University of Colorado at Boulder, USA, 1993.
- [160] M. de Graaf, D. P. Donovan, and A. Apituley, “Saharan desert dust microphysical properties from principle component analysis (PCA) inversion of raman lidar data over western Europe,” in *Proceedings of the 25th International Lidar Radar Conference*, St. Petersburg, Russia, 2010, pp. 671–674.
- [161] M. Castillo-Effer, D. H. Quintela, W. Moreno, R. Jordan, and W. Westhoff, “Wireless sensor networks for flash-flood alerting,” in *Proceedings of the Fifth*

- IEEE International Caracas Conference on Devices, Circuits and Systems, 2004.*, 2004, vol. 1, pp. 142–146.
- [162] T. Gao, D. Greenspan, M. Welsh, R. Juang, and A. Alm, “Vital signs monitoring and patient tracking over a wireless network,” in *Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society, 2005 (IEEE-EMBS 2005)*, 2005, pp. 102–105.
- [163] K. Lorincz, D. J. Malan, T. R. . Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton, “Sensor networks for emergency response: Challenges and opportunities,” *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 16–23, 2004.
- [164] G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton, “Sensor network-based countersniper system,” in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, 2004, pp. 1–12.
- [165] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees, “Deploying a wireless sensor network on an active volcano,” *IEEE Internet Computing*, pp. 18–25, 2006.
- [166] J. Yick, B. Mukherjee, and D. Ghosal, “Analysis of a prediction-based mobility adaptive tracking algorithm,” in *Proceedings of the 2nd International Conference on Broadband Networks, 2005 (BroadNets 2005)*, 2005, pp. 753–760.
- [167] Z. Elabdin, R. Islam, O. O. Khalifa, H. E. . Raouf, and M. J. . Salami, “Development of mathematical model for the prediction of microwave signal

- attenuation due to duststorm,” in *Proceedings of the International Conference on Computer and Communication Engineering (ICCCE 2008)*, 2008, pp. 1156–1161.
- [168] MODIS, “MODIS Website,” 1999. [Online]. Available: <http://modis.gsfc.nasa.gov/data/>. [Accessed: 31-Jan-2012].
- [169] AVHRR, “NOAASIS - NOAA Satellite Information System for NOAA Meteorological,” 1978. [Online]. Available: <http://noaasis.noaa.gov/NOAASIS/ml/avhrr.html>. [Accessed: 31-Jan-2012].
- [170] GOES, “Home Page - NOAA GEOSTATIONARY SATELLITE SERVER,” 1994. [Online]. Available: <http://www.goes.noaa.gov/>. [Accessed: 31-Jan-2012].
- [171] EUMETSAT, “EUMETSAT - Satellites - Instruments,” 2004. [Online]. Available: <http://www.eumetsat.int/Home/Main/Satellites/MeteosatSecondGeneration/Instruments/index.htm?l=en>. [Accessed: 31-Jan-2012].
- [172] S. Chien, B. Cichy, A. Davies, D. Tran, G. Rabideau, R. Castano, R. Sherwood, D. Mandl, S. Frye, S. Shulman, and others, “An autonomous earth-observing sensorweb,” *Intelligent Systems, IEEE*, vol. 20, no. 3, pp. 16–24, 2005.
- [173] H. M. El-Askary, S. Sarkar, M. Kafatos, and T. A. El-Ghazawi, “A multisensor approach to dust storm monitoring over the Nile Delta,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 10, pp. 2386–2391, 2003.
- [174] J. H. Shirley and R. W. Fairbridge, Eds., “Encyclopedia of planetary sciences.” Springer, p. 990, 1997.
- [175] S. A. Ackerman, “Remote sensing aerosols using satellite infrared observations,” *Journal of Geophysical Research*, vol. 102, no. D14, pp. 17069–17, 1997.

- [176] R. C. Allen Jr, P. A. Durkee, and C. H. Wash, “Snow-cloud discrimination with multispectral satellite measurements.,” *Journal of Applied Meteorology*, vol. 29, pp. 994–1004, 1990.
- [177] S. Janugani, V. Jayaram, S. D. Cabrera, J. G. Rosiles, T. E. Gill, and N. R. Rivera, “Directional analysis and filtering for dust storm detection in NOAA-AVHRR imagery,” in *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE)*, 2009, vol. 7334, p. 73341G – 73341G–12.
- [178] K. Wei, T. Zhang, and B. He, “Detection of Sand and Dust Storms from MERIS Image Using FE-Otsu Alogrithm,” in *Proceedings of the 2nd International Conference on Bioinformatics and Biomedical Engineering (ICBBE 2008)*, 2008, pp. 3852–3855.
- [179] M. D. King, Y. J. Kaufman, D. Tanré, and T. Nakajima, “Remote sensing of tropospheric aerosols from space: Past, present, and future,” *Bulletin of the American Meteorological Society*, vol. 80, no. 11, pp. 2229–2259, 1999.
- [180] N. C. Hsu, S. C. Tsay, M. D. King, and J. R. Herman, “Aerosol properties over bright-reflecting source regions,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 3, pp. 557–569, 2004.
- [181] Y. J. Kaufman, D. Tanré, L. Remer, E. Vermote, A. Chu, and B. N. Holben, “Remote sensing of tropospheric aerosol from EOS-MODIS over the land using dark targets and dynamic aerosol models,” *Journal of Geophysical Research*, vol. 102, pp. 17–051, 1997.

- [182] Y. J. Kaufman, A. Wald, L. A. Remer, B. C. Gao, R. R. Li, and L. Flynn, "Remote sensing of aerosol over the continents with the aid of a 2.2 μ m channel," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, pp. 1286–1298, 1997.
- [183] S. Hong, "Detection of Asian dust (Hwangsa) over the Yellow Sea by decomposition of unpolarized infrared reflectivity," *Atmospheric Environment*, vol. 43, no. 37, pp. 5887–5893, 2009.
- [184] V. Tramutoli, C. Filizzola, F. Marchese, G. Mazzeo, R. Paciello, N. Pergola, C. Pietrapertosa, and F. Sannazzaro, "A Robust Satellite Technique (RST) for dust storm detection and monitoring: The case of 2009 Australian event," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2010, pp. 1707–1709.
- [185] H. El-Askary, A. Agarwal, T. El-Ghazawi, M. Kafatos, and J. Le Moigne, "Enhancing dust storm detection using PCA based data fusion," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2005, vol. 2, pp. 1424–1427.
- [186] A. Agarwal, H. M. El-Askary, T. El-Ghazawi, M. Kafatos, and J. Le-Moigne, "Hierarchical PCA Techniques for Fusing Spatial and Spectral Observations With Application to MISR and Monitoring Dust Storms," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 4, pp. 678–682, 2007.
- [187] A. Al Suwaidi, A. Al Rais, and H. Ghedira, "Developing a satellite-based tool to monitor dust and sand storms in the UAE," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Honolulu, HI, 2010, pp. 1434–1437.

- [188] H. Ghedira, A. Al Rais, and A. Al Suwaidi, “Developing a new automated tool for detecting and monitoring dust and sand storms using MODIS and meteosat SEVIRI-MSG data,” in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Cape Town, 2009, vol. 4, p. IV–905 – IV–908.
- [189] P. Zhang, N. Lu, X. Hu, and C. Dong, “Identification and physical retrieval of dust storm using three MODIS thermal IR channels,” *Global and Planetary Change*, vol. 52, no. 1–4, pp. 197–206, 2006.
- [190] J. Ge, J. Huang, F. Weng, and W. Sun, “Effects of dust storms on microwave radiation based on satellite observation and model simulation over the Taklamakan desert,” *Atmospheric Chemistry and Physics Discussions*, vol. 8, no. 2, pp. 7931–7951, 2008.
- [191] P. Rivas-Perea, J. Rosiles, M. Murguia, and J. Tilton, “Automatic Dust Storm Detection Based on Supervised Classification of Multispectral Data,” *Soft Computing for Recognition Based on Biometrics*, pp. 443–454, 2010.
- [192] A. Rashki, D. Kaskaoutis, C. Rautenbach, P. Eriksson, M. Qiang, and P. Gupta, “Dust storms and their horizontal dust loading in the Sistan region, Iran,” *Aeolian Research*, 2012.
- [193] D. G. Kaskaoutis, P. G. Kosmopoulos, P. T. Nastos, H. D. Kambezidis, M. Sharma, and W. Mehdi, “Transport pathways of Sahara dust over Athens, Greece as detected by MODIS and TOMS,” *Geomatics, Natural Hazards and Risk*, vol. 3, no. 1, pp. 35–54, 2011.

- [194] P. Rivas-Perea, J. Rosiles, M. Murguia, and J. Tilton, "Automatic Dust Storm Detection Based on Supervised Classification of Multispectral Data," *Soft Computing for Recognition Based on Biometrics*, pp. 443–454, 2010.
- [195] J. Berni, P. J. Zarco-Tejada, L. Suárez, and E. Fereres, "Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 722–738, 2009.
- [196] A. S. Laliberte, J. E. Herrick, A. Rango, and C. Winters, "Acquisition, orthorectification, and object-based classification of unmanned aerial vehicle (UAV) imagery for rangeland monitoring," *Photogrammetric Engineering and Remote Sensing*, vol. 76, no. 6, pp. 661–672, 2010.
- [197] A. S. Laliberte and A. Rango, "Texture and Scale in Object-Based Analysis of Subdecimeter Resolution Unmanned Aerial Vehicle (UAV) Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 761–770, Mar. 2009.
- [198] A. S. Laliberte, C. Winters, and A. Rango, "A procedure for orthorectification of sub-decimeter resolution imagery obtained with an unmanned aerial vehicle (UAV)," in *Proceedings of the ASPRS 2008 Annual Convention*, Portland, OR, 2008, vol. 28.
- [199] G. Zhou, "Near real-time orthorectification and mosaic of small UAV video flow for time-critical event response," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 739–747, 2009.

- [200] A. Rango, A. Laliberte, J. E. Herrick, C. Winters, K. Havstad, C. Steele, and D. Browning, "Unmanned aerial vehicle-based remote sensing for rangeland assessment, monitoring, and management," *Journal of Applied Remote Sensing*, vol. 3, p. 033542, 2009.
- [201] V. Ramanathan, "Western Pacific Autonomous UAV Campaign: Aerosol-Dust-Cloud Interactions and Climate Effects," *White Paper, Scripps Institution of Oceanography, University of California, San Diego*, p. 28, Oct. 2006.
- [202] A. E. MacDonald, "A global profiling system for improved weather and climate prediction," *Bulletin of the American Meteorological Society*, vol. 86, no. 12, pp. 1747–1764, 2005.
- [203] L. E. Cordova-Lopez, A. Mason, J. D. Cullen, A. Shaw, and A. I. Al-Shamma'a, "Online vehicle and atmospheric pollution monitoring using GIS and wireless sensor networks," in *Journal of Physics: Conference Series*, 2007, vol. 76, p. 012019.
- [204] D. L. Mills, "Internet time synchronization: The network time protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.
- [205] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, pp. c1 –269, 2008.
- [206] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 147–163, 2002.

- [207] S. Ganeriwal, R. Kumar, and M. B. Srivastava, “Timing-sync protocol for sensor networks,” in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, 2003, pp. 138–149.
- [208] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, “The flooding time synchronization protocol,” in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, New York, USA, 2004, pp. 39–49.
- [209] P. Ferrari, A. Flammini, D. Marioli, and A. Taroni, “IEEE 1588-Based Synchronization System for a Displacement Sensor Network,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 57, no. 2, pp. 254–260, Feb. 2008.
- [210] R. L. Scheiterer, C. Na, D. Obradovic, and G. Steindl, “Synchronization Performance of the Precision Time Protocol in Industrial Automation Networks,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 58, no. 6, pp. 1849–1857, Jun. 2009.
- [211] T. Cooklev, J. C. Eidson, and A. Pakdaman, “An Implementation of IEEE 1588 Over IEEE 802.11b for Synchronization of Wireless Local Area Network Nodes,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 56, no. 5, pp. 1632–1639, Oct. 2007.
- [212] D. Cox, E. Jovanov, and A. Milenkovic, “Time synchronization for ZigBee networks,” in *System Theory, 2005. SSST'05. Proceedings of the Thirty-Seventh Southeastern Symposium on*, pp. 135–138.
- [213] M. Aoun, A. Schoofs, and P. van der Stok, “Efficient time synchronization for wireless sensor networks in an industrial setting,” in *Proceedings of the 6th ACM*

Conference on Embedded Network Sensor Systems (SenSys '08), New York, NY, USA, 2008, pp. 419–420.

- [214] M. Akhlaq and T. R. Sheltami, “The Recursive Time Synchronization Protocol for Wireless Sensor Networks,” in *Sensors Applications Symposium (SAS), 2012 IEEE*, Brescia, Italy, 2012, pp. 1–6.
- [215] H. Kopetz and W. Ochsenreiter, “Clock synchronization in distributed real-time systems,” *Computers, IEEE Transactions on*, vol. 100, no. 8, pp. 933–940, 1987.
- [216] Y. Wang, G. Attebury, and B. Ramamurthy, “A survey of security issues in wireless sensor networks,” *IEEE Communications Surveys & Tutorials*, vol. 8, no. 2, pp. 2–23, 2006.
- [217] M. Manzo, T. Roosta, and S. Sastry, “Time synchronization attacks in sensor networks,” in *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, 2005, pp. 107–116.
- [218] V. C. Giruka, M. Singhal, J. Royalty, and S. Varanasi, “Security in wireless sensor networks,” *Wireless communications and mobile computing*, vol. 8, no. 1, pp. 1–24, 2008.
- [219] H. Zhang and J. C. Hou, “Maintaining sensing coverage and connectivity in large sensor networks,” *Ad Hoc & Sensor Wireless Networks*, vol. 1, no. 1–2, pp. 89–124, 2005.
- [220] B. Wang, C. Fu, and H. B. Lim, “Layered diffusion-based coverage control in wireless sensor networks,” *Computer Networks*, vol. 53, no. 7, pp. 1114–1124, 2009.

- [221] Y. Zhou and M. Medidi, "Sleep-based topology control for wakeup scheduling in wireless sensor networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on*, 2007, pp. 304–313.
- [222] P. M. Wightman and M. A. Labrador, "A family of simple distributed minimum connected dominating set-based topology construction algorithms," *Journal of Network and Computer Applications*, 2011.
- [223] S. Jauregui-Ortiz, M. Siller, and F. Ramos, "Node localization in WSN using trigonometric figures," in *Wireless Sensors and Sensor Networks (WiSNet), 2011 IEEE Topical Conference on*, 2011, pp. 65–68.
- [224] K. Srinivasan and P. Levis, "RSSI is Under Appreciated," in *In Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets, Cambridge, MA, 2006*.
- [225] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *Wireless Communications, IEEE Transactions on*, vol. 1, no. 4, pp. 660–670, 2002.
- [226] "Meteosat 0 degree Dust Eastern Africa." [Online]. Available: <http://oiswww.eumetsat.org/IPPS/html/MSG/RGB/DUST/EASTERNAFRICA/>. [Accessed: 27-Jan-2012].
- [227] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007.

- [228] T. R. Sheltami, E. M. Shakshuki, and H. T. Mouftah, "Performance evaluation of TELOSB sensor network," in *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia*, 2009, pp. 584–588.
- [229] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid—A framework for video transmission and quality evaluation," *Computer Performance Evaluation. Modelling Techniques and Tools*, pp. 255–272, 2003.
- [230] C. H. Ke, C. H. Lin, C. K. Shieh, and W. S. Hwang, "A novel realistic simulation tool for video transmission over wireless network," in *Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006. IEEE International Conference on*, 2006, vol. 1, p. 7–pp.
- [231] "EvalVid - A Video Quality Evaluation Tool-set." [Online]. Available: <http://www.tkn.tu-berlin.de/menue/research/evalvid/>. [Accessed: 12-Feb-2012].
- [232] C. Perkins, E. Belding-Royer, S. Das, and others, "RFC 3561-ad hoc on-demand distance vector (AODV) routing," *IETF*, Jul. 2003.
- [233] "The Network Simulator ns2 Homepage." [Online]. Available: <http://www.isi.edu/nsnam/ns/>. [Accessed: 12-Feb-2012].
- [234] T. Schmid, R. Shea, Z. Charbiwala, J. Friedman, M. B. Srivastava, and Y. H. Cho, "On the interaction of clocks, power, and synchronization in duty-cycled embedded sensor nodes," *ACM Transactions on Sensor Networks (TOSN)*, vol. 7, no. 3, p. 24, 2010.

VITA

Muhammad Akhlaq was born in Mandi Bahauddin, Pakistan in 1976. He received his B.Sc. M.Sc and M.Phil degrees in Computer Science from University of the Punjab (PU), Lahore, Pakistan, in 1996, 2001 and 2005 respectively. He has also received Master of Ubiquitous Computing from Blekinge Institute of Technology (BTH), Karlskrona, Sweden in 2010. He has been working as Lecturer-B and PhD student in College of Computer Science & Engineering at King Fahd University of Petroleum & Minerals (KFUPM), Dhahran, Saudi Arabia since 2006. He completed his PhD in Computer Science and Engineering from KFUPM in 2012.

Mr. Muhammad Akhlaq has more than ten years of teaching and research experience in different organizations. He has worked in different projects funded by KFUPM, KACST, PITB, etc. He has published more than thirteen peer-reviewed journal and conference papers. He is also an author of a textbook. His research interests include wireless ad hoc and sensor networks, ubiquitous computing, and context-aware computing.

Mr. Muhammad Akhlaq is a citizen of Pakistan and can be contacted at:

Permanent Address:

House#78/29, Colony Street # 3, Kot
Ahmad Shah, Mandi Bahauddin 50400,
PAKISTAN.
Phone: +92-333-4283856
Email: MuhammadAkhlaq@gmail.com

Current Contact::

ICS department, King Fahd University
of Petroleum & Minerals, Dhahran,
31261, SAUDI ARABIA.
Phone: +966-56-0056471.
Email: akhlaq@kfupm.edu.sa